Download the free study tools at
www.sybex.com/go/ccnaicnd2

Todd Lammle

CCNA®
ICND2
STUDY GUIDE

EXAM 200-101

Includes Real-World Scenarios, Hands-On Exercises, and Access to Exam Prep Software Featuring:

- Free Network Simulator
- Practice Test Environment
- Hundreds of Sample Questions
- Author Videos

SYBEX
A Wiley Brand

CCNA®
ICND2
Study Guide

CCNA®
ICND2
Study Guide

Todd Lammle

SYBEX®
A Wiley Brand

Senior Acquisitions Editor: Jeff Kellum
Development Editor: Pete Gaughan
Technical Editors: John Swartz and Dax Michelson
Production Editor: Christine O'Connor
Copy Editor: Judy Flynn
Editorial Manager: Pete Gaughan
Production Manager: Tim Tate
Vice President and Executive Group Publisher: Richard Swadley
Associate Publisher, Sybex: Chris Webb
Media Project Manager I: Laura Moss-Hollister
Media Associate Producer: Marilyn Hummel
Media Quality Assurance: Doug Kuhn
Book Designers: Judy Fung and Bill Gibson
Compositor: Craig Woods, Happenstance Type-O-Rama
Proofreader: Sarah Kaikini, Word One New York
Indexer: Robert Swanson
Project Coordinator, Cover: Katherine Crocker
Cover Designer: Ryan Sneed

Copyright © 2013 by John Wiley & Sons, Inc., Indianapolis, Indiana

Published simultaneously in Canada

ISBN: 978-1-118-74958-6 (ebk.)

ISBN: 978-1-118-75813-7 (ebk.)

No part of this publication may be reproduced, stored in a retrieval system or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, scanning or otherwise, except as permitted under Sections 107 or 108 of the 1976 United States Copyright Act, without either the prior written permission of the Publisher, or authorization through payment of the appropriate per-copy fee to the Copyright Clearance Center, 222 Rosewood Drive, Danvers, MA 01923, (978) 750-8400, fax (978) 646-8600. Requests to the Publisher for permission should be addressed to the Permissions Department, John Wiley & Sons, Inc., 111 River Street, Hoboken, NJ 07030, (201) 748-6011, fax (201) 748-6008, or online at `http://www.wiley.com/go/permissions`.

Limit of Liability/Disclaimer of Warranty: The publisher and the author make no representations or warranties with respect to the accuracy or completeness of the contents of this work and specifically disclaim all warranties, including without limitation warranties of fitness for a particular purpose. No warranty may be created or extended by sales or promotional materials. The advice and strategies contained herein may not be suitable for every situation. This work is sold with the understanding that the publisher is not engaged in rendering legal, accounting, or other professional services. If professional assistance is required, the services of a competent professional person should be sought. Neither the publisher nor the author shall be liable for damages arising herefrom. The fact that an organization or Web site is referred to in this work as a citation and/or a potential source of further information does not mean that the author or the publisher endorses the information the organization or Web site may provide or recommendations it may make. Further, readers should be aware that Internet Web sites listed in this work may have changed or disappeared between when this work was written and when it is read.

For general information on our other products and services or to obtain technical support, please contact our Customer Care Department within the U.S. at (877) 762-2974, outside the U.S. at (317) 572-3993 or fax (317) 572-4002.

Wiley publishes in a variety of print and electronic formats and by print-on-demand. Some material included with standard print versions of this book may not be included in e-books or in print-on-demand. If this book refers to media such as a CD or DVD that is not included in the version you purchased, you may download this material at `http://booksupport.wiley.com`. For more information about Wiley products, visit `www.wiley.com`.

TRADEMARKS: Wiley, the Wiley logo, and the Sybex logo are trademarks or registered trademarks of John Wiley & Sons, Inc. and/or its affiliates, in the United States and other countries, and may not be used without written permission. CCNA is a registered trademark of Cisco Technology, Inc. All other trademarks are the property of their respective owners. John Wiley & Sons, Inc. is not associated with any product or vendor mentioned in this book.

10 9 8 7 6 5 4 3 2 1

Dear Reader,

Thank you for choosing *CCNA ICND2 Study Guide*. This book is part of a family of premium-quality Sybex books, all of which are written by outstanding authors who combine practical experience with a gift for teaching.

Sybex was founded in 1976. More than 30 years later, we're still committed to producing consistently exceptional books. With each of our titles, we're working hard to set a new standard for the industry. From the paper we print on, to the authors we work with, our goal is to bring you the best books available.

I hope you see all that reflected in these pages. I'd be very interested to hear your comments and get your feedback on how we're doing. Feel free to let me know what you think about this or any other Sybex book by sending me an email at `contactus@sybex.com`. If you think you've found a technical error in this book, please visit `http://sybex.custhelp.com`. Customer feedback is critical to our efforts at Sybex.

Best regards,

Chris Webb
Associate Publisher,
Sybex

Acknowledgments

There are many people who work to put a book together, and although as an author I dedicate an enormous amount of time to write the book, it would never be published without the dedicated, hard work of those other people.

Monica Worthy Lammle was crucial to the finished product this book became, going over every word of every chapter to fine-tune the language and grammar. Without Monica's support in all areas of my life, I'd never have finished writing this book.

Next in line to thank was my new technical editor, John Swartz, who also co-authored the *CCNA Data Center Study Guide* with me. His expertise in the Cisco technical field, and his history of networking in general, is second to none. His detailed analysis of my work helped make this my best CCNA R/S Study Guide ever. Thank you, John, for working hard under pressure, with tight deadlines, and for staying the course of delivering high-quality work in a short time frame.

Jeff Kellum, my acquisitions editor, is instrumental to my success in the Cisco world. Jeff, I look forward to our continued progress together in the Cisco certification world.

Christine O'Connor, my production editor, and Judy Flynn, my copy editor, are my rock and foundation for formatting and intense editing of every page in this book. This amazing team gives me the confidence to help me keep moving during the difficult and very long days, week after week. How Christine stays so organized with all my enormous amount of changes, as well as making sure every figure is in the right place in the book is still a mystery to me! You're amazing, Christine! Thank you! Judy understands my writing style so well now, after doing at least a dozen books with me, that she can tell when I made a technical error that I may have missed when previously going through the material. Thank you, Judy, for doing such a great job! I truly thank you both.

Troy McMillian, Jim Frey, and Paul Sutton really provided support during the writing process: Troy helping me together the bonus questions and flash cards and his super detailed eye when it comes to second, third, and fourth edits, Jim with the amazing figures and all-around administrative support, and Paul with the all-around third-look technical edit. Thank you all!

Finally, a big thanks to Craig Woods at Happenstance-Type-O-Rama and to the Sybex media-development team.

About the Author

Todd Lammle is the authority on Cisco Certification and internetworking and is Cisco certified in many Cisco certification categories. He is a world-renowned author, speaker, trainer, and consultant. Todd has three decades of experience working with LANs, WANs, and large enterprise licensed and unlicensed wireless networks, and lately he's been implementing large Cisco data centers worldwide. His years of real-world experience is evident in his writing; he is not just an author but an experienced networking engineer with very practical experience working on the largest networks in the world at such companies as Xerox, Hughes Aircraft, Texaco, AAA, Cisco, and Toshiba, among many others. Todd has published over 60 books, including the very popular *CCNA: Cisco Certified Network Associate Study Guide*, *CCNA Wireless Study Guide*, and *CCNA Data Center Study Guide*, all from Sybex. He runs an international consulting and training company based in Colorado, Texas, and San Francisco.

You can reach Todd through his forum and blog at `www.lammle.com`.

Contents at a Glance

Introduction *xvii*

ICND2 Exam Objectives *xxvii*

Assessment Test *xxxi*

Chapter 1 Enhanced Switched Technologies 1

Chapter 2 Managing Cisco Devices 47

Chapter 3 IP Services 85

Chapter 4 Troubleshooting IP, IPv6, and VLANs 127

Chapter 5 Enhanced IGRP 169

Chapter 6 Multi-Area OSPF 233

Chapter 7 Wide Area Networks 283

Appendix A Answers to Written Labs 353

Appendix B Answers to Review Questions 359

Appendix C About the Additional Study Tools 375

Index *379*

Contents

Introduction *xvii*

ICND2 Exam Objectives *xxvii*

Assessment Test *xxxi*

Chapter 1 Enhanced Switched Technologies 1

Spanning Tree Protocol (STP) 2
  Spanning-tree Terms 3
  Spanning-tree Operations 7
Types of Spanning-tree Protocols 9
  Common Spanning Tree 10
  Per-VLAN Spanning Tree+ 11
Modifying and Verifying the Bridge ID 17
Spanning-tree Failure Consequences 22
PortFast and BPDU Guard 24
  BPDU Guard 26
EtherChannel 28
  Configuring and Verifying Port Channels 29
Summary 32
Exam Essentials 33
Written Lab 1 34
Hands-on Labs 34
  Hands-on Lab 1.1: Verifying STP and Finding Your Root Bridge 35
  Hands-on Lab 1.2: Configuring and Verifying Your Root Bridge 37
  Hands-on Lab 1.3: Configuring PortFast and BPDU Guard 38
  Hands-on Lab 1.4: Configuring and Verifying EtherChannel 39
Review Questions 42

Chapter 2 Managing Cisco Devices 47

The Internal Components of a Cisco Router 48
The Router Boot Sequence 49
Managing Configuration Register 50
  Understanding the Configuration Register Bits 51
  Checking the Current Configuration Register Value 52
  Boot System Commands 53
  Recovering Passwords 55

Backing Up and Restoring the Cisco IOS 57
  Verifying Flash Memory 59
  Backing Up the Cisco IOS 60
  Restoring or Upgrading the Cisco Router IOS 61
  Using the Cisco IOS File System (Cisco IFS) 63
Licensing 68
  Right-To-Use Licenses (Evaluation Licenses) 70
  Backing Up and Uninstalling the License 73
Summary 74
Exam Essentials 75
Written Labs 77
  Written Lab 2.1: IOS Management 77
  Written Lab 2.2: Router Memory 77
Hands-on Labs 78
  Hands-on Lab 2.1: Backing Up Your Router IOS 78
  Hands-on Lab 2.2: Upgrading or Restoring Your Router IOS 78
Review Questions 80

Chapter 3 IP Services 85

Client Redundancy Issues 86
Introducing First Hop Redundancy Protocol (FHRP) 88
Hot Standby Router Protocol (HSRP) 90
  Virtual MAC Address 91
  HSRP Timers 92
  Group Roles 94
  Configuring and Verifying HSRP 95
Virtual Router Redundancy Protocol 101
  Comparing VRRP and HSRP 101
  VRRP Redundancy Characteristics 101
Gateway Load Balancing Protocol 102
  GLBP Functions 103
  GLBP Features 103
  GLBP Per-host Traffic Balancing 104
  Configuring GLBP 104
Syslog 107
  Configuring and Verifying Syslog 109
SNMP 112
  Management Information Base (MIB) 113
  Configuring SNMP 114
NetFlow 116
  NetFlow Overview and Flows 117
  Configuring NetFlow 118

Summary 120
Exam Essentials 120
Written Lab 3 121
Review Questions 122

Chapter 4 Troubleshooting IP, IPv6, and VLANs 127

Troubleshooting IP Network Connectivity 128
Troubleshooting IPv6 Network Connectivity 140
    ICMPv6 141
Troubleshooting VLAN Connectivity 149
    VLAN Troubleshooting 149
    Trunk Troubleshooting 154
Summary 162
Exam Essentials 163
Written Lab 4 164
Review Questions 165

Chapter 5 Enhanced IGRP 169

EIGRP Features and Operations 170
    Neighbor Discovery 171
    Reliable Transport Protocol (RTP) 176
    Diffusing Update Algorithm (DUAL) 177
    Route Discovery and Maintenance 178
Configuring EIGRP 178
    VLSM Support and Summarization 181
    Controlling EIGRP Traffic 184
    Split Horizon 195
Verifying and Troubleshooting EIGRP 197
    Troubleshooting Example with EIGRP 204
EIGRPv6 214
Summary 219
Exam Essentials 220
Written Lab 5 221
Hands-on Labs 221
    Hands-on Lab 5.1: Configuring and Verifying EIGRP 221
    Hands-on Lab 5.2: Configuring and Verifying EIGRPv6 223
Review Questions 224

Chapter 6 Multi-Area OSPF 233

OSPF Scalability 234
Categories of Multi-area Components 236
    Adjacency Requirements 236
    OSPF Router Roles 237

Link-state Advertisements 238
OSPF Hello Protocol 240
Neighbor States 241
Basic Multi-area Configuration 243
Verifying and Troubleshooting Multi-area OSPF Networks 245
The *show ip ospf* Command 247
The *show ip ospf interface* Command 248
The *show ip protocols* Command 250
The *show ip route* Command 251
The *show ip ospf database* Command 252
Troubleshooting OSPF Scenario 254
OSPFv3 262
Verifying OSPFv3 265
Summary 267
Exam Essentials 267
Written Lab 6 269
Hands-on Labs 269
Hands-on Lab 6.1: Configuring and Verifying OSPF Multi-Area 270
Hands-on Lab 6.2: Configuring and Verifying OSPFv3 273
Review Questions 276

Chapter 7 Wide Area Networks 283

Introduction to Wide Area Networks 284
Defining WAN Terms 285
WAN Connection Bandwidth 286
WAN Connection Types 286
WAN Support 288
Cable and DSL 291
Cable 292
Digital Subscriber Line (DSL) 293
Cabling the Serial Wide Area Network 296
Serial Transmission 296
Data Terminal Equipment and Data Communication Equipment 297
High-Level Data-Link Control (HDLC) Protocol 298
Point-to-Point Protocol (PPP) 300
Link Control Protocol (LCP) Configuration Options 301
PPP Session Establishment 302
PPP Authentication Methods 303
Configuring PPP on Cisco Routers 303
Configuring PPP Authentication 303
Verifying and Troubleshooting Serial Links 304

Frame Relay 309
    Introduction to Frame Relay Technology 310
    Frame Relay Implementation and Monitoring 317
Virtual Private Networks 328
    Benefits of VPNs 329
    Introduction to Cisco IOS IPsec 330
    IPsec Transforms 331
GRE Tunnels 332
    Configuring GRE Tunnels 333
    Verifying GRP Tunnels 335
Summary 337
Exam Essentials 337
Written Lab 7 339
Hands-on Labs 339
    Hands-on Lab 7.1: Configuring PPP Encapsulation and Authentication 339
    Hands-on Lab 7.2: Configuring and Monitoring HDLC 341
    Hands-on Lab 7.3: Configuring Frame Relay and Subinterfaces 342
    Hands-on Lab 7.4: Configuring a GRE Tunnel 344
Review Questions 347

Appendix A Answers to Written Labs 353

Chapter 1: Enhanced Switch Technologies 354
Chapter 2: Managing Cisco Devices 354
    Written Lab 2.1: IOS Management 354
    Written Lab 2.2: Router Memory 355
Chapter 3: IP Services 355
Chapter 4: Troubleshooting IP, IPv6, and VLANs 356
Chapter 5: Enhanced IGRP 356
Chapter 6: Multi-Area OSPF 356
Chapter 7: Wide Area Networks 357

Appendix B Answers to Review Questions 359

Chapter 1: Enhanced Switch Technologies 360
Chapter 2: Managing Cisco Devices 362
Chapter 3: IP Services 364
Chapter 4: Troubleshooting IP, IPv6, and VLANs 367
Chapter 5: Enhanced IGRP 368
Chapter 6: Multi-Area OSPF 370
Chapter 7: Wide Area Networks 372

Appendix C About the Additional Study Tools 375

Additional Study Tools 376
  Test Engine 376
  Electronic Flashcards 376
  Videos 376
  Network Simulator 377
  PDF of Glossary of Terms 377
  Adobe Reader 377
System Requirements 377
Using the Study Tools 377
Troubleshooting 378
  Customer Care 378

Index 379

Introduction

Welcome to the exciting world of Cisco certification! If you've picked up this book because you want to improve yourself and your life with a better, more satisfying and secure job, you've done the right thing. Whether you're striving to enter the thriving, dynamic IT sector or seeking to enhance your skill set and advance your position within it, being Cisco certified can seriously stack the odds in your favor to help you attain your goals!

Cisco certifications are powerful instruments of success that also markedly improve your grasp of all things internetworking. As you progress through this book, you'll gain a complete understanding of networking that reaches far beyond Cisco devices. By the end of your studies, you'll comprehensively know how disparate network topologies and technologies work together to form the fully operational networks that are vital to today's very way of life in the developed world. The knowledge and expertise you'll gain here is essential for and relevant to every networking job and is why Cisco certifications are in such high demand—even at companies with few Cisco devices!

Although it's now common knowledge that Cisco rules routing and switching, the fact that it also rocks the voice, data center, and service provider world is also well recognized. And Cisco certifications reach way beyond the popular but less extensive certifications like those offered by CompTIA and Microsoft to equip you with indispensable insight into today's vastly complex networking realm. Essentially, by deciding to become Cisco certified, you're proudly announcing that you want to become an unrivaled networking expert—a goal that this book will get you well on your way to achieving. Congratulations in advance on the beginning of your brilliant future!

NOTE

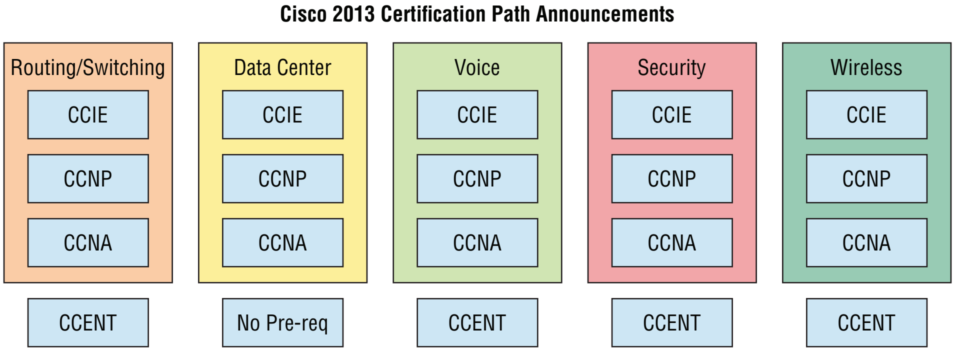
For up-to-the-minute updates covering additions or modifications to the Cisco certification exams, as well as additional study tools, review questions, and bonus material, be sure to visit the Todd Lammle websites and forum at `www.lammle.com`, `www.lammlesim.com`, and `www.lammle.com/forum`.

Cisco's Network Certifications

It used to be that to secure the holy grail of Cisco certifications—the CCIE—you passed only one written test before being faced with a grueling, formidable hands-on lab. This intensely daunting, all-or-nothing approach made it nearly impossible to succeed and predictably didn't work out too well for most people. Cisco responded to this issue by creating a series of new certifications, which not only made it easier to eventually win the highly coveted CCIE prize, it gave employers a way to accurately rate and measure the skill levels of prospective and current employees. This exciting paradigm shift in Cisco's certification path truly opened doors that few were allowed through before!

Beginning in 1998, obtaining the Cisco Certified Network Associate (CCNA) certification was the first milestone in the Cisco certification climb, as well as the unofficial prerequisite to each of the more advanced levels. But that changed when Cisco announced the Cisco Certified Entry Network Technician (CCENT) certification. And then in March 2013, Cisco once again proclaimed that all-new tests will be required beginning in October of the same year; now the Cisco certification process looks like Figure I.1.

FIGURE I.1 The Cisco certification path



NOTE

I have included only the most popular tracks in Figure I.1. In addition to the ones in this image, there are also tracks for Design, Service Provider, Service Provider Operations, and Video. Also note that the CCIE Voice certification retirement will be announced shortly.

The Cisco Routing and Switching (R/S) path is by far the most popular and could very well remain so, but soon you'll see the Data Center path become more and more of a focus as companies migrate to data center technologies. The Voice track also actually does provide a good job opportunity. Still, understanding the foundation of R/S before attempting any other certification track is something I highly recommend.

Even so, and as the figure shows, you only need your CCENT certification to get underway for most of the tracks. Also, note that there are a few other certification tracks you can go down that are not shown in the figure, although they're not as popular as the ones shown. You can find information on all Cisco certification tracks at `www.cisco.com`.

Cisco Certified Entry Network Technician (CCENT)

Don't be fooled by the oh-so-misleading name of this first certification because it absolutely isn't entry level! Okay—maybe entry level for Cisco's certification path, but definitely not for someone without experience trying to break into the highly lucrative yet challenging IT job market! For the uninitiated, the CompTIA A+ and Network+ certifications aren't official prerequisites, but know that Cisco does expect you to have that type and level of experience before embarking on your Cisco certification journey.

All of this gets us to 2013, when the climb to Cisco supremacy just got much harder again. The innocuous-sounding siren's call of the CCENT can lure you to some serious trouble if you're not prepared, because it's actually much harder than the old CCNA ever was. This will rapidly become apparent once you start studying, but be encouraged! The fact that the certification process is getting harder really works better for you in the long run, because that which is harder to obtain only becomes that much more valuable when you finally do, right? Yes, indeed!

Another important factor to keep in mind is that the CCENT 100-101 exam, which is the exam this book was written for, costs $150 per attempt and it's anything but easy to pass! The good news is that this book will guide you step-by-step in building a strong foundation in routing and switching technologies. You really need to build on a strong technical foundation and stay away from exam cram type books, suspicious online material, and the like. They can help somewhat, but understand that you'll pass the Cisco certification exams only if you have a strong foundation and that you'll get that solid foundation only by reading as much as you can, performing the written labs and review questions in this book, and practicing lots and lots of hands-on labs. Additional practice exam questions, videos, and labs are offered on my website, and what seems like a million other sites offer additional material that can help you study.

However, there is one way to skip the CCENT exam and still meet the prerequisite before moving on to any other certification track, and that path is through the CCNA R/S composite exam. First, I'll discuss the Interconnecting Cisco Network Devices 2 (ICND2) exam, and then tell you about the composite CCNA exam, which will provide you, when successful, with both the CCENT and the CCNA R/S certification.

Cisco Certified Network Associate Routing and Switching (CCNA R/S)

Once you have achieved your CCENT certification, you can take the ICND2 (200-101) exam in order to achieve your CCNA R/S certification. This is now the most popular certification Cisco has by far because it's the most sought-after certification of all employers.

As with the CCENT, this exam is also $150 per attempt—although thinking you can just skim a book and pass any of these exams would probably be a really expensive mistake! The CCENT/CCNA exams are extremely hard and cover a lot of material, so you have to really know your stuff. Taking a Cisco class or spending months with hands-on experience is definitely a requirement to succeed when faced with this monster!

And once you have your CCNA, you don't have to stop there—you can choose to continue and achieve an even higher certification, called the Cisco Certified Network Professional (CCNP). There are various ones, as shown in Figure I.1. The CCNP R/S is still the most popular, with Voice certifications coming in at a close second. And I've got to tell you that the Data Center certification will be catching up fast. Also good to know is that anyone with a CCNP R/S has all the skills and knowledge needed to attempt the notoriously dreaded but coveted CCIE R/S lab. But just becoming a CCNA R/S can land you that job you've dreamed about, and that's what this book is all about: helping you to get and keep a great job!

Still, why take two exams to get your CCNA if you don't have to? Cisco still has the composite exam called CCNA 200-120 that, if passed, will land you with your CCENT and your CCNA R/S via only one test priced accordingly at $300. Some people like the one-test approach, and some people like the two-test approach; this ICND2 book will help you with the two-test method.

Why Become a CCENT and CCNA R/S?

Cisco, like Microsoft and other vendors that provide certification, has created the certification process to give administrators a set of skills and to equip prospective employers with a way to measure those skills or match certain criteria. And as you probably know, becoming a CCNA R/S is certainly the initial, key step on a successful journey toward a new, highly rewarding, and sustainable networking career.

The CCNA program was created to provide a solid introduction not only to the Cisco Internetwork Operating System (IOS) and Cisco hardware but also to internetworking in general, making it helpful to you in areas that are not exclusively Cisco's. And regarding today's certification process, it's not unrealistic that network managers—even those without Cisco equipment—require Cisco certifications for their job applicants.

Rest assured that if you make it through the CCNA and are still interested in Cisco and internetworking, you're headed down a path to certain success!

What Skills Do You Need to Become a CCENT?

This ICND1 exam tests a candidate for the knowledge and skills required to successfully install, operate, and troubleshoot a small branch office network.

The exam includes questions on the operation of IP data networks, LAN switching technologies, IPv6, IP routing technologies, IP services network device security, and basic troubleshooting.

What Skills Do You Need to Become a CCNA R/S After You Have Passed ICND1?

The ICND2 exam (exam 200-101) tests a candidate for the knowledge and skills required to successfully install, operate, and troubleshoot a small- to medium-size enterprise branch network. The exam includes questions on LAN switching technologies, IP routing technologies, IP services (FHRP, syslog, SNMP v2 and v3), troubleshooting, and WAN technologies.

Another test you can take instead of taking both of the ICND1 and ICND2 is the composite CCNA v2.0 exam, number 200-120.

How Do You Become a CCNA R/S with Just One Test?

The way to become a CCNA R/S is to pass one little test (CCNA composite exam 200-120) without having to take the CCENT exam. Then—poof!—you're a CCNA R/S. Oh, but don't you wish it were that easy? True, it's just one test, but it's a whopper, and to pass it you must possess enough knowledge to understand what the test writers are saying and you need to know everything I mentioned in the ICND1 and ICND2 sections above! Hey, it's hard, but it can be done!

What does it cover: the CCNA Exam: 200-120? CCNA exam is the composite exam associated with the CCNA Routing and Switching certification. Candidates can prepare for this exam by taking the Todd Lammle authorized Cisco boot camps. This composite exam tests a candidate's knowledge and skills required to install, operate, and troubleshoot a small- to medium-size enterprise branch network. The topics include all of the areas covered under the 100-101 ICND1 and 200-101 ICND2 exams.

You can take the one composite test (200-120), but it is good to remember that Cisco offers the two-step process to become a CCNA as I discussed earlier in this Introduction. It may be easier than taking that one ginormous exam for you, but don't think the two-test method is easy. It takes work! However, it can be done; you just need to stick with your studies.The two-test method involves passing the following:

- Exam 100-101: Interconnecting Cisco Networking Devices Part 1 (ICND1)
- Exam 200-101: Interconnecting Cisco Networking Devices Part 2 (ICND2)

I can't stress this point enough: It's critical that you have some hands-on experience with Cisco routers. If you can get ahold of some basic routers and switches, you're set, but if you can't, I've worked hard to provide hundreds of configuration examples throughout this book to help network administrators, or people who want to become network administrators, learn the skills they need to pass the CCENT and CCNA R/S exams. In addition, a simulator called LammleSim IOS version is available for free with the purchase of this book. This small simulator will run through all the hands-on labs found in this book—nice huh?

NOTE

For Cisco certification hands-on training with CCSI Todd Lammle, please see `www.lammle.com`. Each student will get hands-on experience by configuring at least three routers and two switches—no sharing of equipment!

What Does This Book Cover?

This book covers everything you need to know to pass the ICND2 200-101 exam. But as I've said, taking plenty of time to study and practice with routers or a router simulator is the real key to success.

You will learn the following information in this book:

Chapter 1: Enhanced Switched Technologies This chapter will start off with STP protocols and dive into the fundamentals, covering modes as well as various flavors of STP. VLANs, trunks, and troubleshooting are covered as well. EtherChannel technologies, configuration, and verification are also covered. There are hands-on labs, a written lab, and plenty of review questions to help you. Do not even think of skipping the fundamental written and hands-on labs in this chapter!

Chapter 2: Managing Cisco Devices This chapter describes the boot process of Cisco routers, the configuration register, and how to manage Cisco IOS files. The chapter finishes with a section on Cisco's new licensing strategy for its IOS. Hands-on and written labs, along with review questions, will help you build a strong foundation for the objectives covered in this chapter.

Chapter 3: IP Services This chapter mostly focuses on first hop redundancy protocols (FHRPs) such as HSRP and GLBP. Also covered is syslog, SNMP, and NetFlow. There are a written lab and review questions.

Chapter 4: Troubleshooting IP, IPV6, and VLANs I want to say this is the most important chapter in the book, but that's hard to say. You can decide that yourself when you take the exam! Be sure to go through all the troubleshooting steps for IP, IPv6 and VLANs. The hands-on labs for this chapter will be included in the free bonus material and be dynamic labs that I'll write and change as needed. Don't skip the written lab and review questions.

Chapter 5: Enhanced EIGRP This is a full chapter on nothing but EIGRP and EIGRPv6. There are lots of examples, including configurations, verification, and troubleshooting labs, with both IP and with IPv6. Great hands-on labs are included, as well as a written lab and review questions.

Chapter 6: Multi-Area OSPF Before reading this chapter, be sure you have the CCENT objectives down pat with a strong OSPF foundation. This chapter will take off where that CCENT OSPF chapter left off, and add multi-area networks along with advanced configurations and then finish with OSPFv3. Hands-on labs, a written lab, and challenging review questions await you at the end of the chapter.

Chapter 7: Wide Area Networks This is the longest, and last, chapter in the book. It covers a multiple of protocols in depth, especially HDLC, PPP, and Frame Relay, along with a discussion on many other technologies. Good troubleshooting examples are provided in the PPP and Frame Relay configuration sections, and these cannot be skipped! Hands-on labs meant to focus squarely on the objectives are included at the end of the chapter, as well as a written lab and challenging review questions.

Appendix A: Answers to Written Labs This appendix contains the answers to the book's written labs.

Appendix B: Answers to Chapter Review Questions This appendix provides the answers to the end-of-chapter review questions.

Appendix C: About the Additional Study Tools This describes the technical requirements for the digital study tools that come with this book. (Those tools are described further in the following section!)

TIP

Be sure to check the announcements section of my forum to find out how to download bonus material I created specifically for this book.

What's Available Online?

I have worked hard to provide some really great tools to help you with your certification process. All of the following tools, most of them available at `www.sybex.com/go/ccnaicnd2`, should be loaded on your workstation when you're studying for the test. As a fantastic bonus, I was able to add to the download link a preview section from my CCNA video series! Please understand that these are not the full versions, but they're still a great value for you included free with this book.

Test Preparation Software The test preparation software prepares you to pass the ICND2 exam. In the test engine, you'll find all the review and assessment questions from the book *plus* additional practice exam questions that appear exclusively among the downloadable additional study tools.

Electronic Flashcards The companion study tools include 100 flashcards specifically written to hit you hard, so don't get discouraged if you don't ace your way through them at first! They're there to ensure that you're really ready for the exam. And no worries—armed with the review questions, practice exams, and flashcards, you'll be more than prepared when exam day comes!

LammleSim IOS Version At `www.lammle.com` or `www.lammlesim.com`, I have provided an IOS simulator that can be used with all of the hands-on labs in this book. This is also included on the book's downloads page at `www.sybex.com/go/ccnaicnd2`.

Glossary A complete glossary of CCENT, CCNA, and Cisco routing terms is available at `www.sybex.com/go/ccnaicnd2`.

Todd Lammle Bonus Material and Labs Be sure to check the announcement section of my forum at `www.lammle.com/forum` for directions on how to download all the latest bonus material created specifically to help you study for your ICND2 and CCNA Routing and Switching exams.

Todd Lammle Videos I have created a full series of CCNA videos that can be purchased in either DVD or downloadable format from `www.lammle.com`. As a bonus included with this book, I've included modules 7, 8, and 9. These can be downloaded from the book's web page, `www.sybex.com/go/ccnaicnd2`, as a preview. Although this isn't the full version, the video included with this book is over 1 hour of foundational CCNA information. This is a $99 value, so don't skip this video because it covers the integral topics on the CCENT/CCNA exams.

CBT Nuggets Videos

In addition, CBT Nuggets has created dozens of free MicroNugget videos that help reinforce the chapter topics. Throughout the book, you will see QR codes and URLs that will direct you to a dedicated website where you can view these videos created by CBT Nuggets author and expert trainer Jeremy Cioara. Just scan the QR code with your smart phone or tablet, or type in the URL, to view the video.

How to Use This Book

If you want a solid foundation for the serious effort of preparing for the Interconnecting Cisco Network Devices 2 (ICND2) 200-101 exam, then look no further. I've spent hundreds of hours putting together this book with the sole intention of helping you to pass the ICND2 exam as well as really learn how to correctly configure Cisco routers and switches!

This book is loaded with valuable information, and you will get the most out of your study time if you understand why the book is organized the way it is.

So to maximize your benefit from this book, I recommend the following study method:

1. Take the assessment test that's provided at the end of this introduction. (The answers are at the end of the test.) It's okay if you don't know any of the answers; that's why you bought this book! Carefully read over the explanations for any question you get wrong and note the chapters in which the material relevant to them is covered. This information should help you plan your study strategy.
2. Study each chapter carefully, making sure you fully understand the information and the test objectives listed at the beginning of each one. Pay extra-close attention to any chapter that includes material covered in questions you missed.
3. Complete the written labs at the end of each chapter. (Answers to these appear in Appendix A.) Do *not* skip these written exercises because they directly relate to the ICND2 exam and what you must glean from the chapters in which they appear. Do not just skim these labs! Make sure you completely understand the reason for each correct answer.

4. Complete all hands-on labs in each chapter, referring to the text of the chapter so that you understand the reason for each step you take. Try to get your hands on some real equipment, but if you don't have Cisco equipment available, try the LammleSim IOS version, which you can use for the hands-on labs found only in this book. These labs will equip you with everything you need for all your Cisco certification goals.
5. Answer all of the review questions related to each chapter. (The answers appear in Appendix B.) Note the questions that confuse you, and study the topics they cover again, until the concepts are crystal clear. And again—do not just skim these questions! Make sure you fully comprehend the reason for each correct answer. Remember that these will not be the exact questions you will find on the exam, but they're written to help you understand the chapter material and ultimately pass the exam!
6. Try your hand at the practice questions that are exclusive to this book. The questions can be found only at `www.sybex.com/go/ccnaicnd2`. And be sure to check out `www.lammle.com` for the most up-to-date Cisco exam prep questions, videos, Todd Lammle boot camps, and more.
7. Also on the download link are modules from my complete CCNA video series.

NOTE

Please understand that these are preview versions of the video and audios found at `www.lammle.com`. The videos on the free download link are not the full versions, but they're still a great value loaded with information.

8. Test yourself using all the flashcards, which are also found on the download link. These are brand-new and updated flashcards to help you prepare for the ICND2 exam and a wonderful study tool!

To learn every bit of the material covered in this book, you'll have to apply yourself regularly, and with discipline. Try to set aside the same time period every day to study, and select a comfortable and quiet place to do so. I'm confident that if you work hard, you'll be surprised at how quickly you learn this material!

If you follow these steps and really study—*doing hands-on labs every single day* in addition to using the review questions, the practice exams, the Todd Lammle video sections, and the electronic flashcards as well as all the written labs—it would actually be hard to fail the ICND2 exam. But understand that studying for the ICND2 exam is a lot like getting in shape—if you do not go to the gym every day, it's not going to happen!

Where Do You Take the Exams?

You may take the ICND2 or any Cisco exam at any of the Pearson VUE authorized testing centers. For information, check `www.vue.com` or call 877-404-EXAM (3926).

To register for a Cisco exam, follow these steps:

1. Determine the number of the exam you want to take. (The ICND2 exam number is 200-101.)
2. Register with the nearest Pearson VUE testing center. At this point, you will be asked to pay in advance for the exam. At the time of this writing, the exam is $150 and must be taken within one year of payment. You can schedule exams up to six weeks in advance or as late as the day you want to take it—but if you fail a Cisco exam, you must wait five days before you will be allowed to retake it. If something comes up and you need to cancel or reschedule your exam appointment, contact Pearson VUE at least 24 hours in advance.
3. When you schedule the exam, you'll get instructions regarding all appointment and cancellation procedures, the ID requirements, and information about the testing-center location.

Tips for Taking Your ICND2 Exam

The ICND2 exam contains about 50 questions and must be completed in 90 minutes or less. This information can change per exam. You must get a score of about 85 percent to pass this exam, but again, each exam can be different.

Many questions on the exam have answer choices that at first glance look identical—especially the syntax questions! So remember to read through the choices carefully because close just doesn't cut it. If you get commands in the wrong order or forget one measly character, you'll get the question wrong. So, to practice, do the hands-on exercises at the end of this book's chapters over and over again until they feel natural to you.

Also, never forget that the right answer is the Cisco answer. In many cases, more than one appropriate answer is presented, but the *correct* answer is the one that Cisco recommends. On the exam, you will always be told to pick one, two, or three options, never "choose all that apply." The ICND2 exam may include the following test formats:

- Multiple-choice single answer
- Multiple-choice multiple answer
- Drag-and-drop
- Router simulations

Cisco proctored exams will not show the steps to follow in completing a router interface configuration, but they do allow partial command responses. For example, `show run` or `sho running` or `sh running-config` would be acceptable.

Here are some general tips for exam success:

- Arrive early at the exam center so you can relax and review your study materials.
- Read the questions *carefully*. Don't jump to conclusions. Make sure you're clear about *exactly* what each question asks. "Read twice, answer once," is what I always tell my students.

- When answering multiple-choice questions that you're not sure about, use the process of elimination to get rid of the obviously incorrect answers first. Doing this greatly improves your odds if you need to make an educated guess.
- You can no longer move forward and backward through the Cisco exams, so double-check your answer before clicking Next since you can't change your mind.

After you complete an exam, you'll get immediate, online notification of your pass or fail status, a printed examination score report that indicates your pass or fail status, and your exam results by section. (The test administrator will give you the printed score report.) Test scores are automatically forwarded to Cisco within five working days after you take the test, so you don't need to send your score to them. If you pass the exam, you'll receive confirmation from Cisco, typically within two to four weeks, sometimes a bit longer.

ICND2 Exam Objectives

Exam objectives are subject to change at any time without prior notice and at Cisco's sole discretion. Please visit Cisco's certification website (`www.cisco.com/web/learning`) for the latest information on the ICND2 exam.

LAN Switching Technologies	**Chapter(s)**
Identify enhanced switching technologies ▪ RSTP ▪ PVSTP ▪ Ethercnannels	1
Configure and verify PVSTP operation ▪ Describe root bridge election ▪ Spanning tree mode	1
IP Routing Technologies	
Describe the boot process of Cisco IOS routers ▪ POST ▪ Router bootup process	2
Configure and verify operation status of a serial interface	7

Manage Cisco IOS Files
▪ Boot preferences
▪ Cisco IOS image(s)
▪ Licensing
    ▪ Show license
    ▪ Change license
	2

LAN Switching Technologies	**Chapter(s)**
Differentiate methods of routing and routing protocols	5
■ Administrative distance	
■ Split horizon	
■ Metric	
■ Next hop	
Configure and verify OSPF (single area)	6
■ Neighbor adjacencies	
■ OSPF states	
■ Discuss Multi-area	
■ Configure OSPF v2	
■ Configure OSPF v3	
■ Router ID	
■ LSA types	
Configure and verify EIGRP (single AS)	5
■ Feasible Distance / Feasible Successors / Administrative Distance	
■ Feasibility condition	
■ Metric composition	
■ Router ID	
■ Auto-summary	
■ Path selection	
■ Load balancing	
■ Equal	
■ Unequal	
■ Passive interface	
IP Services	
Recognize High availability (FHRP)	3
■ VRRP	
■ HSRP	
■ GLBP	
Configure and verify Syslog	3
■ Utilize Syslog Output	
Describe SNMP v2 & v3	3

LAN Switching Technologies	Chapter(s)
Troubleshooting	
Identify and correct common network problems	4
Utilize NetFlow data	3
Troubleshoot and resolve spanning-tree operation issues ▪ Root switch ▪ Priority ▪ Mode is correct ▪ Port states	1
Troubleshoot and resolve routing issues ▪ Routing is enabled ▪ Routing table is correct ▪ Correct path selection	4, 5, 6
Troubleshoot and resolve OSPF problems ▪ Neighbor adjacencies ▪ Hello and Dead timers ▪ OSPF area ▪ Interface MTU ▪ Network types ▪ Neighbor states ▪ OSPF topology database	6
Troubleshoot and resolve EIGRP problems ▪ Neighbor adjacencies ▪ AS number ▪ Load balancing ▪ Split horizon	5
Troubleshoot and resolve interVLAN routing problems ▪ Connectivity ▪ Encapsulation ▪ Subnet ▪ Native VLAN ▪ Port mode trunk status	4

LAN Switching Technologies	Chapter(s)
Troubleshoot and resolve WAN implementation issues - Serial interfaces - PPP - Frame relay	7
Monitor NetFlow statistics	3
Troubleshoot EtherChannel problems	1
WAN Technologies	
Identify different WAN Technologies - Metro Ethernet - VSAT - Cellular 3G/4G - MPLS - T1/E1 - ISDN - DSL - Frame relay - Cable - VPN	7
Configure and verify a basic WAN serial connection	7
Configure and verify a PPP connection between Cisco routers	7
Configure and verify frame relay on Cisco routers	7
Implement and troubleshoot PPPoE	7

Assessment Test

1. What is the `sys-id-ext` field in a BPDU used for?
   - **A.** This is a 4-bit field inserted into an Ethernet frame to define trunking information between switches.
   - **B.** This is a 12-bit field inserted into an Ethernet frame to define VLANs in an STP instance.
   - **C.** This is a 4-bit field inserted into an non-Ethernet frame to define EtherChannel options.
   - **D.** This is a 12-bit field inserted into an Ethernet frame to define STP root bridges.

2. You have four RSTP PVST+ links between switches and want to aggregate the bandwidth. What solution will you use?
   - **A.** EtherChannel
   - **B.** PortFast
   - **C.** BPDU Channel
   - **D.** VLANs
   - **E.** EtherBundle

3. What configuration parameters must be configured the same between switches for LACP to form a channel? (Choose three.)
   - **A.** Virtual MAC address
   - **B.** Port speeds
   - **C.** Duplex
   - **D.** PortFast enabled
   - **E.** VLAN information

4. You reload a router with a configuration register setting of 0x2101. What will the router do when it reloads?
   - **A.** The router enters setup mode.
   - **B.** The router enters ROM monitor mode.
   - **C.** The router boots the mini-IOS in ROM.
   - **D.** The router expands the first IOS in flash memory into RAM.

5. Which of the following commands provides the product ID and serial number of a router?

A. `show license`

B. `show license feature`

C. `show version`

D. `show license udi`

6. Which command allows you to view the technology options and licenses that are supported on your router along with several status variables?

A. `show license`

B. `show license feature`

C. `show license udi`

D. `show version`

7. Which of the following will you use to collect details about network traffic patterns on your network, including protocols used?

A. `SNMPv3`

B. `Syslogv2`

C. `NetFlow 9`

D. `logging host` *`ip_address`*

8. You want to send console message to a syslog server, but you only want to send status messages of 3 and lower. Which of the following commands will you use?

A. `logging trap emergencies`

B. `logging trap errors`

C. `logging trap debugging`

D. `logging trap notifications`

E. `logging trap critical`

F. `logging trap warnings`

G. `logging trap alerts`

9. When is the AVR used in GLBP?

A. When HSRP preempts GLBP

B. When a client needs the virtual MAC

C. When the FHRP is set to passive mode and load balancing

D. When the FHRP is load-balancing between VLANs with VVRP

E. When there is a redundancy failure

10. You need to connect to a remote IPv6 server in your virtual server farm. You can connect to the IPv4 servers, but not the critical IPv6 server you desperately need. Based on the output, what could your problem be?

```
C:\>ipconfig
   Connection-specific DNS Suffix  . : localdomain
   IPv6 Address. . . . . . . . . . . : 2001:db8:3c4d:3:ac3b:2ef:1823:8938
   Temporary IPv6 Address. . . . . . : 2001:db8:3c4d:3:2f33:44dd:211:1c3d
   Link-local IPv6 Address . . . . . : fe80::ac3b:2ef:1823:8938%11
   IPv4 Address. . . . . . . . . . . : 10.1.1.10
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 10.1.1.1
```

A. The global address is in the wrong subnet.

B. The IPv6 default gateway has not been configured or received from the router.

C. The link-local address has not been resolved so the host cannot communicate to the router.

D. There are two IPv6 global addresses configured. One must be removed from the configuration.

11. What command is used to view the IPv6-to-MAC-address resolution table on a Cisco router?

A. `show ip arp`

B. `show ipv6 arp`

C. `show ip neighbors`

D. `show ipv6 neighbors`

E. `show arp`

12. An IPv6 ARP entry is listed with a status of REACH. What can you conclude about the IPv6-to-MAC-address mapping?

A. The interface has communicated with the neighbor address and the mapping is current.

B. The interface has not communicated within the neighbor reachable time frame.

C. The ARP entry has timed out.

D. IPv6 can reach the neighbor address but the addresses has not yet been resolved.

13. Serial0/1 goes down. How will EIGRP send packets to the 10.1.1.0 network?

```
Corp#show ip eigrp topology
[output cut]
P 10.1.1.0/24, 2 successors, FD is 2681842
         via 10.1.2.2 (2681842/2169856), Serial0/0
         via 10.1.3.1 (2973467/2579243), Serial0/2
         via 10.1.3.3 (2681842/2169856), Serial0/1
```

A. EIGRP will put the 10.1.1.0 network into active mode.
B. EIGRP will drop all packets destined for 10.1.1.0.
C. EIGRP will just keep sending packets out s0/0.
E. EIGRP will use s0/2 as the successor and keep routing to 10.1.1.0.

14. What command produced the following output?

```
via FE80::201:C9FF:FED0:3301 (29110112/33316), Serial0/0/0
via FE80::209:7CFF:FE51:B401 (4470112/42216), Serial0/0/1
via FE80::209:7CFF:FE51:B401 (2170112/2816), Serial0/0/2
```

A. `show ip protocols`
B. `show ipv6 protocols`
C. `show ip eigrp neighbors`
D. `show ipv6 eigrp neighbors`
E. `show ip eigrp topology`
F. `show ipv6 eigrp topology`

15. You need to troubleshoot an adjacency between two EIGRP configured routers. What should you look for? (Choose four.)

A. Verify the AS numbers.
B. Verify that you have the proper interfaces enabled for EIGRP.
C. Make sure there are no mismatched K-values.
D. Check your passive interface settings.
E. Make sure your remote routers are not connected to the Internet.
F. If authentication is configured, make sure all routers use different passwords.

16. You have two OSPF directly configured routers that are not forming an adjacency. What should you check? (Choose three.)
   - **A.** Process ID
   - **B.** Hello and dead timers
   - **C.** Link cost
   - **D.** Area
   - **E.** IP address/subnet mask

17. When do two adjacent routers enter the 2WAY state?
   - **A.** After both routers have received Hello information
   - **B.** After they have exchanged topology databases
   - **C.** When they connect only to a DR or BDR
   - **D.** When they need to exchange RID information

18. Which type of LSAs are generated by ABRs and referred to as summary link advertisements (SLAs)?
   - **A.** Type 1
   - **B.** Type 2
   - **C.** Type 3
   - **D.** Type 4
   - **E.** Type 5

19. Which of the following is *not* provided by the AH portion of IPsec?
   - **A.** Integrity
   - **B.** Confidentiality
   - **C.** Authenticity
   - **D.** Anti-replay

20. Which statement about GRE is not true?
   - **A.** GRE is stateless and has no flow control.
   - **B.** GRE has security.
   - **C.** GRE has additional overhead for tunneled packets, at least 24 bytes.
   - **D.** GRE uses a protocol-type field in the GRE header so any layer 3 protocol can be used through the tunnel.

21. How should a router that is being used in a frame relay network be configured to keep split horizon issues from preventing routing updates?

A. Configure a single subinterface to establish multiple PVC connections to multiple remote router interfaces.

B. Combine multiple frame relay circuits as a point-to-point line to support multicast and broadcast traffic.

C. Configure many subinterfaces in the same subnet.

D. Configure a separate subinterface for each PVC with a unique DLCI and subnet assigned to the subinterface.

Answers to Assessment Test

1. B. To allow for the PVST+ to operate, there's a field inserted into the BPDU to accommodate the extended system ID so that PVST+ can have a root bridge configured on a per-STP instance. The extended system ID (VLAN ID) is a 12-bit field, and we can even see what this field is carrying via show spanning-tree command output. See Chapter 1 for more information.
2. A. Cisco's EtherChannel can bundle up to eight ports between switches in order to provide resiliency and more bandwidth between switches. See Chapter 1 for more information.
3. B, C, E. All the ports on both sides of every link must be configured exactly the same between switches or EtherChannel will not work. Speed, duplex, and allowed VLANs must match. See Chapter 1 for more information.
4. C. 2100 boots the router into ROM monitor mode, 2101 load the mini-IOS from ROM, 2102 is the default and loads the IOS from flash. See Chapter 2 for more information.
5. D. The `show license udi` command displays the unique device identifier (UDI) of the router, which comprises the product ID (PID) and serial number of the router. See Chapter 2 for more information.
6. B. The `show license` feature command allows you to view the technology package licenses and feature licenses that are supported on your router along with several status variables related to software activation and licensing, both licensed and unlicensed features. See Chapter 2 for more information.
7. C. NetFlow allows the collection of flows of traffic on the network, including address and protocols in use. A recent version of NetFlow is version 9, referred to as Flexible NetFlow. See Chapter 3 for more information.
8. B. There are eight different trap levels: If you choose, for example, level 3, levels 0 through 3 will be displayed. See Chapter 3 for more information.
9. B. The active virtual router (AVR) responds with the virtual MAC addresses to the clients. See Chapter 3 for more information.
10. B. There is no IPv6 default gateway, which will be the link-local address of the router interface, sent to the host as a router advertisement. Until this host receives the router address, the host will communicate with IPv6 only on the local subnet. See Chapter 4 for more information.
11. D. The command `show ipv6 neighbors` provides the ARP cache on a router. See Chapter 4 for more information.

12. A. The state is STALE when the interface has not communicated within the neighbor reachable time frame. The next time the neighbor communicates, the state will be REACH. See Chapter 4 for more information.

13. C. There are two successor routes, so by default, EIGRP was load-balancing out s0/0 and s0/1. When s0/1 goes down, EIGRP will just keep forwarding traffic out the second link, s0/0. S0/1 will be removed from the routing table. See Chapter 5 for more information.

14. F. There isn't a lot to go on with the output, but the only commands that provide the FD and AD is show ip eigrp topology and show ipv6 eigrp topology. The addresses in the output are link-local IPv6 addresses, so our answer is the latter. See Chapter 5 for more information.

15. A, B, C, D. Cisco has documented steps, according to the objectives, that you must go through when troubleshooting an adjacency. See Chapter 5 for more information.

16. B, D, E. In order for two OSPF routers to create an adjacency, the hello and dead timers must match, and they must both be configured into the same area, as well as being in the same subnet. See Chapter 6 for more information.

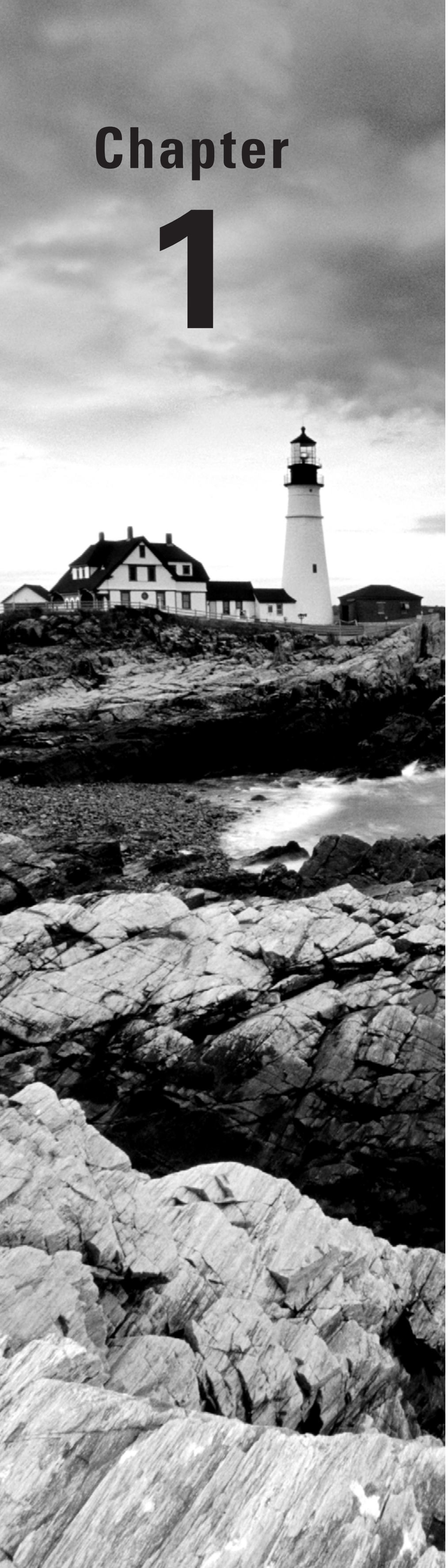
17. A. The process starts by sending out Hello packets. Every listening router will then add the originating router to the neighbor database. The responding routers will reply with all of their Hello information so that the originating router can add them to its own neighbor table. At this point, we will have reached the 2WAY state—only certain routers will advance beyond this. See Chapter 6 for more information.

18. C. Referred to as summary link advertisements (SLAs), Type 3 LSAs are generated by area border routers. These ABRs send Type 3 LSAs toward the area external to the one where they were generated. See Chapter 6 for more information.

19. B. AH checks the entire packet, but it doesn't offer any encryption services. See Chapter 7 for more information.

20. B. GRE is a generic tunnel protocol that has no built-in security. The rest of the options are correct for GRE tunnels. See Chapter 7 for more information.

21. D. If you have a serial port configured with multiple DLCIs connected to multiple remote sites, split horizon rules stop route updates received on an interface from being sent out the same interface. By creating subinterfaces for each PVC, you can avoid the split horizon issues when using frame relay. See Chapter 7 for more information.



Chapter 1

Enhanced Switched Technologies

THE FOLLOWING ICND2 EXAM TOPICS ARE COVERED IN THIS CHAPTER:

✓ **LAN Switching Technologies**

- Identify enhanced switching technologies
  - RSTP
  - PVSTP
  - EtherChannels
- Configure and verify PVSTP operation
  - describe root bridge election
  - spanning-tree mode

✓ **Troubleshooting**

- Troubleshoot and resolve spanning-tree operation issues
  - root switch
  - priority
  - mode is correct
  - port states
- Troubleshoot EtherChannel problems

Long ago, a company called Digital Equipment Corporation (DEC) created the original version of *Spanning Tree Protocol (STP)*. The IEEE later created its own version of STP called 802.1d. Cisco has moved toward another industry standard in its newer switches called 802.1w. We'll explore both the old and new versions of STP in this chapter, but first, I'll define some important STP basics.

Routing protocols like RIP and OSPF have processes for preventing loops from occurring at the Network layer, but if you have redundant physical links between your switches, these protocols won't do a thing to stop loops from occurring at the Data Link layer. That's exactly why STP was developed—to put an end to loop issues in a layer 2 switched network. It's also why we'll be thoroughly exploring the key features of this vital protocol as well as how it works within a switched network in this chapter.

After covering STP in detail, we'll move on to explore EtherChannel.

NOTE

To find up-to-the-minute updates for this chapter, please see `www.lammle.com/forum` or the book's web page at `www.sybex.com`.

Spanning Tree Protocol (STP)

Spanning Tree Protocol (STP) achieves its primary objective of preventing network loops on layer 2 network bridges or switches by monitoring the network to track all links and shut down the redundant ones. STP uses the spanning-tree algorithm (STA) to first create a topology database and then search out and disable redundant links. With STP running, frames will be forwarded on only premium, STP-chosen links.

The Spanning Tree Protocol is a great protocol to use in networks like the one shown in Figure 1.1.

This is a switched network with a redundant topology that includes switching loops. Without some type of layer 2 mechanism in place to prevent a network loop, this network is vulnerable to nasty issues like broadcast storms, multiple frame copies, and MAC table thrashing! Figure 1.2 shows how this network would work with STP working on the switches.

There a few types of spanning-tree protocols, but I'll start with the IEEE version 802.1d, which happens to be the default on all Cisco IOS switches.

FIGURE 1.1 A switched network with switching loops

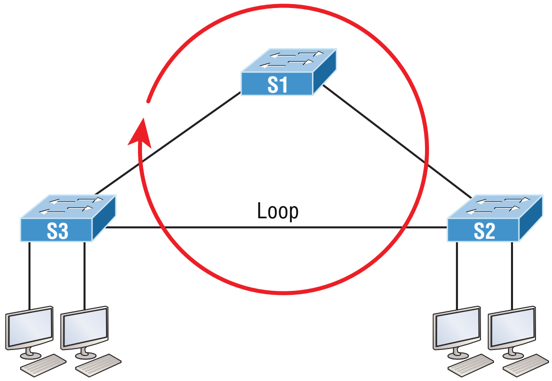
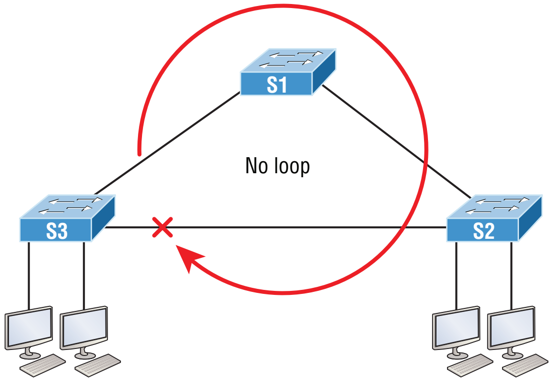


FIGURE 1.2 A switched network with STP



Spanning-tree Terms

Now, before I get into describing the details of how STP works within a network, it would be good for you to have these basic ideas and terms down first:

Root bridge The *root bridge* is the bridge with the lowest and, therefore, the best bridge ID. The switches within the STP network elect a root bridge, which becomes the focal point in the network. All other decisions in the network, like which ports on the non root bridges should be blocked or put in forwarding mode, are made from the perspective of the root bridge, and once it has been elected, all other bridges must create a single path to it. The port with the best path to the root bridge is called the root port.

Non-root bridges These are all bridges that aren't the root bridge. Non-root bridges exchange BPDUs with all the other bridges and update the STP topology database on all switches. This prevents loops and helps defend against link failures.

BPDU All switches exchange information to use for the subsequent configuration of the network. Each switch compares the parameters in the *Bridge Protocol Data Unit (BPDU)* that it sends to a neighbor with the parameters in the BPDU that it receives from other neighbors. Inside the BPDU is the bridge ID.

Bridge ID The bridge ID is how STP keeps track of all the switches in the network. It's determined by a combination of the bridge priority, which is 32,768 by default on all Cisco switches, and the base MAC address. The bridge with the lowest bridge ID becomes the root bridge in the network. Once the root bridge is established, every other switch must make a single path to it. Most networks benefit by forcing a specific bridge or switch to be on the Root Bridge by setting its bridge priority lower than the default value.

Port cost Port cost determines the best path when multiple links are used between two switches. The cost of a link is determined by the bandwidth of a link, and this path cost is the deciding factor used by every bridge to find the most efficient path to the root bridge.

Path cost A switch may encounter one or more switches on its path to the Root Bridge, and there may be more than one possible path. All unique paths are analyzed individually, and a path cost is calculated for each unique path by adding the individual port costs encountered on the way to the Root Bridge.

Bridge Port Roles

STP uses roles to determine how a port on a switch will act within the spanning-tree algorithm.

Root port The root port is the link with the lowest path cost to the root bridge. If more than one link connects to the root bridge, then a port cost is found by checking the bandwidth of each link. The lowest-cost port becomes the root port. When multiple links connect to the same device, the port connected to the lowest port number on the upstream switch will be the one that's used. The root bridge can never have a root port designation, while every other switch in a network must have one and only one root port.

Designated port A *designated port* is one that's been determined to have the best (lowest) cost to get to on a given network segment, compared to other ports on that segment. A designated port will be marked as a forwarding port, and you can have only one forwarding port per network segment.

Non-designated port A *non-designated port* is one with a higher cost than the designated port. These are basically the ones left over after the root ports and designated ports have been determined. Non-designated ports are put in blocking or discarding mode—they are not forwarding ports!

Forwarding port A forwarding port forwards frames and will be either a root port or a designated port.

Blocked port A blocked port won't forward frames in order to prevent loops. A blocked port will still always listen to BPDU frames from neighbor switches, but it will drop any and all other frames received and will never transmit a frame.

Alternate port This corresponds to the blocking state of 802.1d, and is a term used with the newer 802.1w (Cisco Rapid Spanning Tree Protocol). An alternative port is located on a switch connected to a LAN segment with two or more switches connected, and one of the other switches holds the designated port.

Backup port This corresponds to the blocking state of 802.1d, and is a term now used with the newer 802.1w. A backup port is connected to a LAN segment where another port on that switch is acting as the designated port.

Spanning-tree Port States

Okay, so you plug your host into a switch port and the light turns amber and your host doesn't get a DHCP address from the server. You wait and wait and finally the light goes green after almost a full minute—that's an eternity in today's networks! This is the STA transitioning through the different port states verifying that you didn't just create a loop with the device you just plugged in. STP would rather time out your new host than allow a loop into the network because that would effectively bring your network to its knees. Let's talk about the transition states; then later in this chapter we'll talk about how to speed this process up.

The ports on a bridge or switch running IEEE 802.1d STP can transition through five different states:

Disabled (technically, not a transition state) A port in the administratively disabled state doesn't participate in frame forwarding or STP. A port in the disabled state is virtually nonoperational.

Blocking As I mentioned, a blocked port won't forward frames; it just listens to BPDUs. The purpose of the blocking state is to prevent the use of looped paths. All ports are in blocking state by default when the switch is powered up.

Listening This port listens to BPDUs to make sure no loops occur on the network before passing data frames. A port in listening state prepares to forward data frames without populating the MAC address table.

Learning The switch port listens to BPDUs and learns all the paths in the switched network. A port in learning state populates the MAC address table but still doesn't forward data frames. Forward delay refers to the time it takes to transition a port from listening to learning mode, or from learning to forwarding mode, which is set to 15 seconds by default and can be seen in the `show spanning-tree` output.

Forwarding This port sends and receives all data frames on the bridged port. If the port is still a designated or root port at the end of the learning state, it will enter the forwarding state.

NOTE

Switches populate the MAC address table in learning and forwarding modes only.

Switch ports are most often in either the blocking or forwarding state. A forwarding port is typically the one that's been determined to have the lowest (best) cost to the root bridge. But when and if the network experiences a topology change due to a failed link or because someone has added in a new switch, you'll see the ports on a switch transitioning through listening and learning states.

As I said earlier, blocking ports is a strategy for preventing network loops. Once a switch determines the best path to the root bridge for its root port and any designated ports, all other redundant ports will be in blocking mode. Blocked ports can still receive BPDUs—they just don't send out any frames.

If a switch determines that a blocked port should become the designated or root port because of a topology change, it will go into listening mode and check all BPDUs it receives to make sure it won't create a loop once the port moves into forwarding mode.

Convergence

Convergence occurs when all ports on bridges and switches have transitioned to either forwarding or blocking modes. No data will be forwarded until convergence is complete. Yes—you read that right: When STP is converging, all host data stops transmitting through the switches! So if you want to remain on speaking terms with your network's users, or remain employed for any length of time, you must make sure that your switched network is physically designed really well so that STP can converge quickly!

Convergence is vital because it ensures that all devices have a coherent database. And making sure this happens efficiently will definitely require your time and attention. The original STP (802.1d) takes 50 seconds to go from blocking to forwarding mode by default and I don't recommend changing the default STP timers. You can adjust those timers for a large network, but the better solution is simply to opt out of using 802.1d at all! We'll get to the various STP versions in a minute.

Link Costs

Now that you know about the different port roles and states, you need to really understand all about path cost before we put this all together. Port cost is based on the speed of the link, and Table 1.1 breaks down the need-to-know path costs for you. Port cost is the cost of a single link whereas path cost is the sum of the various port costs to the root bridge.

TABLE 1.1 IEEE STP link costs

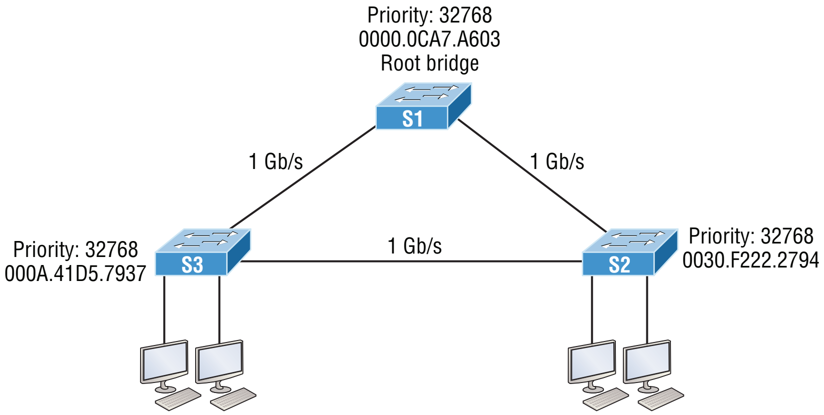
Speed	Cost
10 Mb/s	100
100 Mb/s	19
1000 Mb/s	4
10,000 Mb/s	2

These costs will be used in the STP calculations to choose a single root port on each bridge. You absolutely need to memorize this table, but no worries—I'll guide you through lots of examples in this chapter to help you do that quite easily! Now it's time to take everything we've learned so far and put it all together.

Spanning-tree Operations

Let's start neatly summarizing what you've learned so far using the simple three-switch network connected together as shown in Figure 1.3.

FIGURE 1.3 STP operations



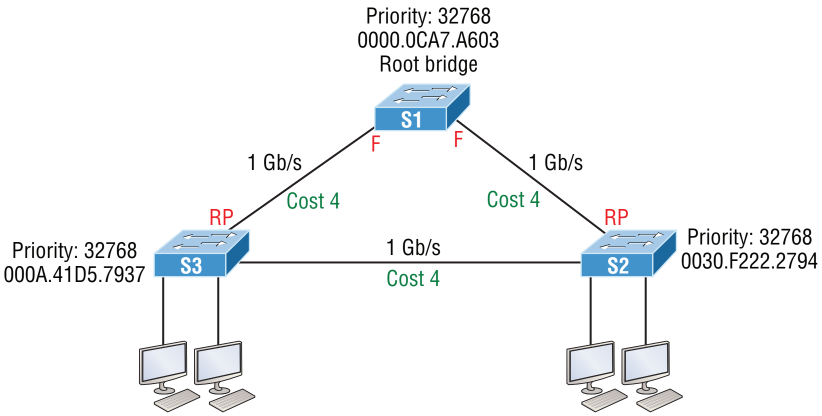
Basically, STP's job is to find all the links in the network and shut down any redundant ones, thereby preventing network loops from occurring. It achieves this by first electing a root bridge that will have all ports forwarding and will also act as a point of reference for all other devices within the STP domain. In Figure 1.4, S1 has been elected the root bridge based on bridge ID. Since the priorities are all equal to 32,768, we'll compare MAC addresses and find the MAC address of S1 is lower than that of S2 and S3, meaning that S1 has a better bridge ID.

Once all switches agree on the root bridge, they must then determine their one and only root port—the single path to the root bridge. It's really important to remember that a bridge can go through many other bridges to get to the root, so it's not always the shortest path that will be chosen. That role will be given to the port that happens to offer the fastest, highest bandwidth. Figure 1.4 shows the root ports for both non-root bridges (the RP signifies a root port and the F signifies a designated forwarding port).

Looking at the cost of each link, it's clear why S2 and S3 are using their directly connected links, because a gigabit link has a cost of 4. For example, if S3 chose the path through S2 as its root port, we'd have to add up each port cost along the way to the root, which would be 4 + 4 for a total cost of 8.

Every port on the root bridge is a designated, or forwarding, port for a segment, and after the dust settles on all other non-root bridges, any port connection between switches that isn't either a root port or a designated port will predictably become a non-designated port. These will again be put into the blocking state to prevent switching loops.

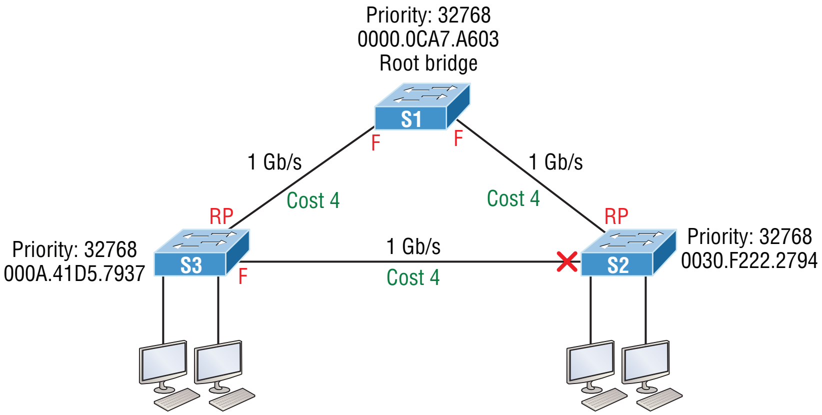
FIGURE 1.4 STP operations


Priority: 32768
0000.0CA7.A603
Root bridge
S1
F
F
1 Gb/s
1 Gb/s
Cost 4
Cost 4
RP
RP
Priority: 32768
000A.41D5.7937
S3
1 Gb/s
Cost 4
S2
Priority: 32768
0030.F222.2794

Okay—at this point, we have our root bridge with all ports in forwarding state and we've found our root ports for each non-root bridge. Now the only thing left to do is to choose the one forwarding port on the segment between S2 and S3. Both bridges can't be forwarding on a segment because that's exactly how we would end up with loops. So, based on the bridge ID, the port with the best and lowest would become the only bridge forwarding on that segment, with the one having the highest, worst bridge ID put into blocking mode. Figure 1.5 shows the network after STP has converged.

Since S3 had a lower bridge ID (better), S2's port went into blocking mode. Let's discuss the root bridge election process more completely now.

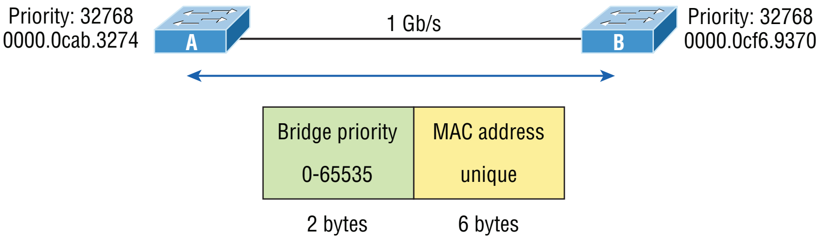
FIGURE 1.5 STP operations


Priority: 32768
0000.0CA7.A603
Root bridge
S1
F
F
1 Gb/s
1 Gb/s
Cost 4
Cost 4
RP
RP
Priority: 32768
000A.41D5.7937
S3
F
1 Gb/s
Cost 4
S2
Priority: 32768
0030.F222.2794

Selecting the Root Bridge

The bridge ID is used to elect the root bridge in the STP domain and to determine the root port for each of the remaining devices when there's more than one potential root port available because they have equal-cost paths. This key bridge ID is 8 bytes long and includes both the priority and the MAC address of the device, as illustrated in Figure 1.6. Remember—the default priority on all devices running the IEEE STP version is 32,768.

FIGURE 1.6 STP operations



So, to determine the root bridge, you combine the priority of each bridge with its MAC address. If two switches or bridges happen to have the same priority value, the MAC address becomes the tiebreaker for figuring out which one has the lowest and, therefore, best ID. This means that because the two switches in Figure 1.6 are both using the default priority of 32,768, the MAC address will be the determining factor instead. And because Switch A's MAC address is 0000.0cab.3274 and Switch B's MAC address is 0000.0cf6.9370, Switch A wins and will become the root bridge. A really easy way to figure out the lowest MAC address is to just start reading from the left toward the right until you find a lesser value. For Switch A, I only needed to get to 0000.0ca before stopping. Switch A wins since switch B is 0000.0cf. Never forget that the lower value is always the better one when it comes to electing a root bridge!

I want to point out that prior to the election of the root bridge, BPDUs are sent every 2 seconds out all active ports on a bridge/switch by default, and they're received and processed by all bridges. The root bridge is elected based on this information. You can change the bridge's ID by lowering its priority so that it will become a root bridge automatically. Being able to do that is important in a large switched network because it ensures that the best paths will actually be the ones chosen. Efficiency is always awesome in networking!

Types of Spanning-tree Protocols

There are several varieties of spanning-tree protocols in use today:

IEEE 802.1d The original standard for bridging and STP, which is really slow but requires very little bridge resources. It's also referred to as Common Spanning Tree (CST).

PVST+ The Cisco proprietary enhancement for STP that provides a separate 802.1d spanning-tree instance for each VLAN. Know that this is just as slow as the CST protocol, but with it, we get to have multiple root bridges. This creates more efficiency of the links in the network, but it does use more bridge resources than CST does.

IEEE 802.1w Also called Rapid Spanning Tree Protocol (RSTP), this iteration enhanced the BPDU exchange and paved the way for much faster network convergence, but it still only allows for one root bridge per network like CST. The bridge resources used with RSTP are higher than CST's but less than PVST+.

Rapid PVST+ Cisco's version of RSTP that also uses PVST+ and provides a separate instance of 802.1w per VLAN. It gives us really fast convergence times and optimal traffic flow but predictably requires the most CPU and memory of all.

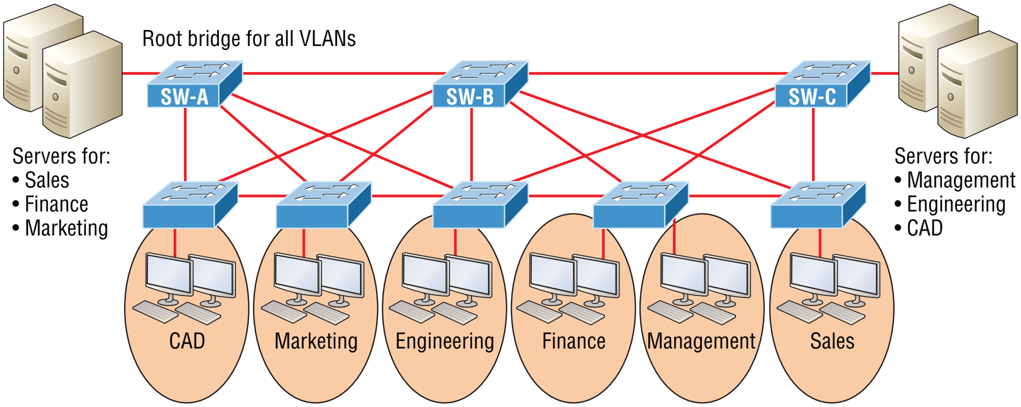
Common Spanning Tree

If you're running Common Spanning Tree (CST) in your switched network with redundant links, there will be an election to choose what STP considers to be the best root bridge for your network. That switch will also become the root for all VLANs in your network and all bridges in your network will create a single path to it. You can manually override this selection and pick whichever bridge you want if it makes sense for your particular network.

Figure 1.7 shows how a typical root bridge would look on your switched network when running CST.

Notice that switch A is the root bridge for all VLANs even though it's really not the best path for some VLANs because all switches must make a single path to it! This is where Per-VLAN Spanning Tree+ (PVST+) comes into play. Because it allows for a separate instance of STP for each VLAN, it frees up the individual selection of the most optimal path.

FIGURE 1.7 Common STP example

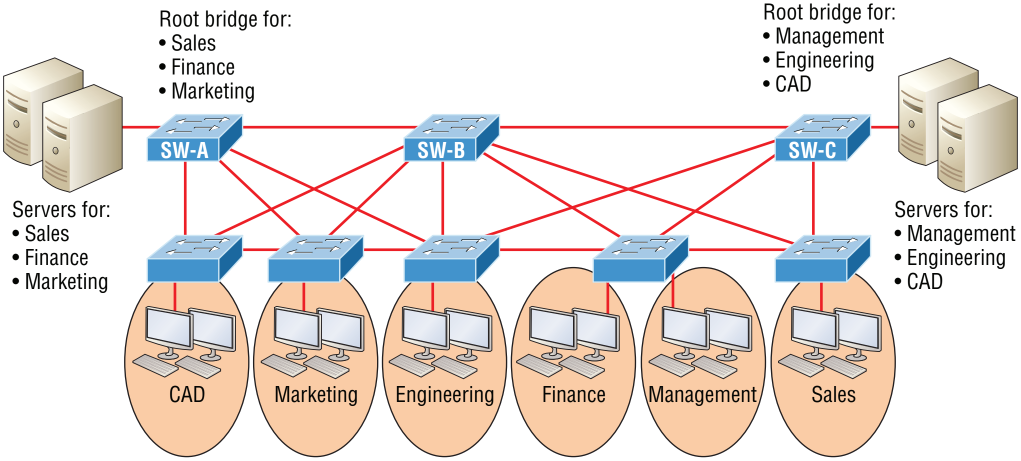


Per-VLAN Spanning Tree+

PVST+ is a Cisco proprietary extension to 801.2d STP that provides a separate 802.1 spanning-tree instance for each VLAN configured on your switches. All of Cisco proprietary extensions were created to improve convergence times, which is 50 seconds by default. Cisco IOS switches run 802.1d PVST+ by default, which means you'll have optimal path selection, but the convergence time will still be slow.

Creating a per-VLAN STP instance for each VLAN is worth the increased CPU and memory requirements, because it allows for per-VLAN root bridges. This feature allows the STP tree to be optimized for the traffic of each VLAN by allowing you to configure the root bridge in the center of each of them. Figure 1.8 shows how PVST+ would look in an optimized switched network with multiple redundant links.

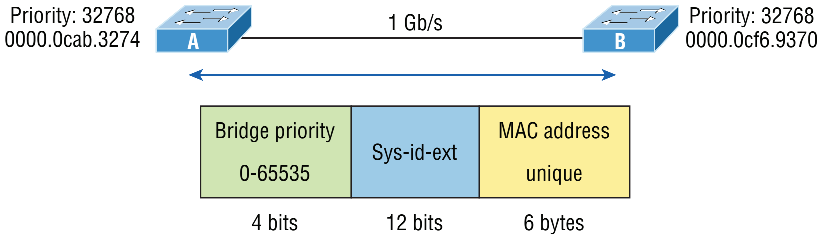
FIGURE 1.8 PVST+ provides efficient root bridge selection.


Root bridge for:
• Sales
• Finance
• Marketing
Root bridge for:
• Management
• Engineering
• CAD
SW-A
SW-B
SW-C
Servers for:
• Sales
• Finance
• Marketing
Servers for:
• Management
• Engineering
• CAD
CAD
Marketing
Engineering
Finance
Management
Sales

This root bridge placement clearly enables faster convergence as well as optimal path determination. This version's convergence is really similar to 802.1 CST's, which has one instance of STP no matter how many VLANs you have configured on your network. The difference is that with PVST+, convergence happens on a per-VLAN basis, with each VLAN running its own instance of STP. Figure 1.8 shows us that we now have a nice, efficient root bridge selection for each VLAN.

To allow for the PVST+ to operate, there's a field inserted into the BPDU to accommodate the extended system ID so that PVST+ can have a root bridge configured on a per-STP instance, shown in Figure 1.9. The bridge ID actually becomes smaller—only 4 bits—which means that we would configure the bridge priority in blocks of 4,096 rather than in increments of 1 as we did with CST. The extended system ID (VLAN ID) is a 12-bit field, and we can even see what this field is carrying via `show spanning-tree` command output, which I'll show you soon.

FIGURE 1.9 PVST+ unique bridge ID



But still, isn't there a way we can do better than a 50-second convergence time? That's a really long time in today's world!

Rapid Spanning Tree Protocol 802.1w

Wouldn't it be wonderful to have a solid STP configuration running on your switched network, regardless of switch type, and still have all the features we just discussed built in and enabled on every one of your switches too? Rapid Spanning Tree Protocol (RSTP) serves up exactly this amazing capacity right to our networking table!

Cisco created proprietary extensions to "fix" all the sinkholes and liabilities the IEEE 802.1d standard threw at us, with the main drawback to them being they require extra configuration because they're Cisco proprietary. But RSTP, the new 802.1w standard, brings us most of the patches needed in one concise solution. Again, efficiency is golden!

RSTP, or IEEE 802.1w, is essentially an evolution of STP that allows for much faster convergence. But even though it does address all the convergence issues, it still only permits a single STP instance, so it doesn't help to take the edge off suboptimal traffic flow issues. And as I mentioned, to support that faster convergence, the CPU usage and memory demands are slightly higher than CST's. The good news is that Cisco IOS can run the Rapid PVST+ protocol—a Cisco enhancement of RSTP that provides a separate 802.1w spanning-tree instance for each VLAN configured within the network. But all that power needs fuel, and although this version addresses both convergence and traffic flow issues, it also demands the most CPU and memory of all solutions. And it's also good news that Cisco's newest switches don't have a problem with this protocol running on them.

NOTE Keep in mind that Cisco documentation may say STP 802.1d and RSTP 802.1w, but it is referring to the PVST+ enhancement of each version.

Understand that RSTP wasn't meant to be something completely new and different. The protocol is more of an evolution than an innovation of the 802.1d standard, which offers faster convergence whenever a topology change occurs. Backward compatibility was a must when 802.1w was created.

So, RSTP helps with convergence issues that were the bane of traditional STP. Rapid PVST+ is based on the 802.1w standard in the same way that PVST+ is based on 802.1d.

The operation of Rapid PVST+ is simply a separate instance of 802.1w for each VLAN. Here's a list to clarify how this all breaks down:

- RSTP speeds the recalculation of the spanning tree when the layer 2 network topology changes.
- It's an IEEE standard that redefines STP port roles, states, and BPDUs.
- RSTP is extremely proactive and very quick, so it doesn't need the 802.1d delay timers.
- RSTP (802.1w) supersedes 802.1d while remaining backward compatible.
- Much of the 802.1d terminology and most parameters remain unchanged.
- 802.1w is capable of reverting to 802.1d to interoperate with traditional switches on a per-port basis.

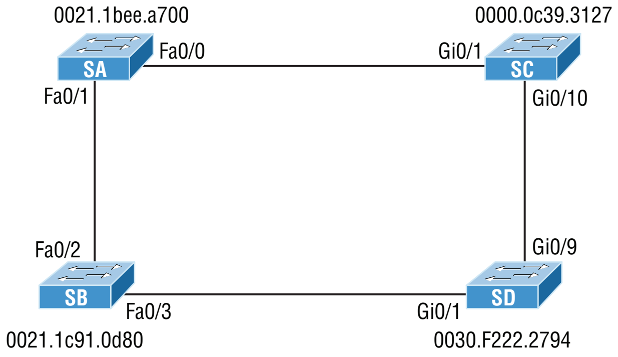
And to clear up confusion, there are also five terminology adjustments between 802.1d's five port states to 802.1w's, compared here, respectively:

802.1d State		**802.1w State**
Disabled	=	Discarding
Blocking	=	Discarding
Listening	=	Discarding
Learning	=	Learning
Forwarding	=	Forwarding

Make note of the fact that RSTP basically just goes from discarding to learning to forwarding, whereas 802.1d requires five states to transition.

The task of determining the root bridge, root ports, and designated ports hasn't changed from 802.1d to RSTP, and understanding the cost of each link is still key to making these decisions well. Let's take a look at an example of how to determine ports using the revised IEEE cost specifications in Figure 1.10.

FIGURE 1.10 RSTP example 1



Can you figure out which is the root bridge? How about which port is the root and which ones are designated? Well, because SC has the lowest MAC address, it becomes the root bridge, and since all ports on a root bridge are forwarding designated ports, well, that's easy, right? Ports Gi0/1 and Gi0/10 become designated forwarding ports on SC.

But which one would be the root port for SA? To figure that out, we must first find the port cost for the direct link between SA and SC. Even though the root bridge (SC) has a Gigabit Ethernet port, it's running at 100 Mbps because SA's port is a 100-Mbps port, giving it a cost of 19. If the paths between SA and SC were both Gigabit Ethernet, their costs would only be 4, but because they're running 100 Mbps links instead, the cost jumps to a whopping 19!

Can you find SD's root port? A quick glance at the link between SC and SD tells us that's a Gigabit Ethernet link with a cost of 4, so the root port for SD would be its GI0/9 port.

The cost of the link between SB and SD is also 19 because it's also a Fast Ethernet link, bringing the full cost from SB to SD to the root (SC) to a total cost of 19 + 4 = 23. If SB were to go through SA to get to SC, then the cost would be 19 + 19, or 38, so the root port of SB becomes the Fa0/3 port.

The root port for SA would be the Fa0/0 port since that's a direct link with a cost of 19. Going through SB to SD would be 19 + 19 + 4 = 42, so we'll use that as a backup link for SA to get to the root just in case we need to.

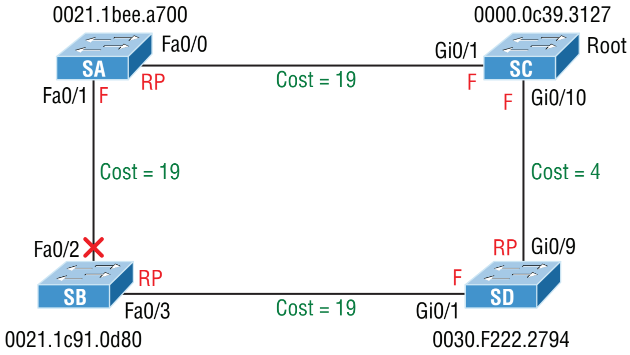
Now, all we need is a forwarding port on the link between SA and SB. Because SA has the lowest bridge ID, Fa0/1 on SA wins that role. Also, the Gi0/1 port on SD would become a designated forwarding port. This is because the SB Fa0/3 port is a designed root port and you must have a forwarding port on a network segment! This leaves us with the Fa0/2 port on SB. Since it isn't a root port or designated forwarding port, it will be placed into blocking mode, which will prevent looks in our network.

Let's take a look at this example network when it has converged in Figure 1.11.

If this isn't clear and still seems confusing, just remember to always tackle this process following these three steps:

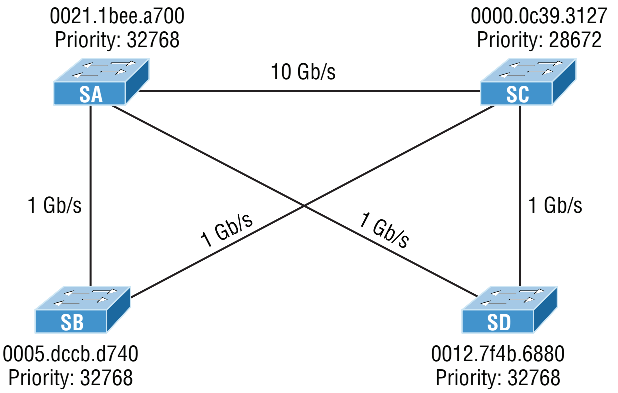
1. Find your root bridge by looking at bridge IDs.
2. Determine your root ports by finding the lowest path cost to the root bridge.
3. Find your designated ports by looking at bridge IDs.

FIGURE 1.11 RSTP example 1 answer



As usual, the best way to nail this down is to practice, so let's explore another scenario, shown in Figure 1.12.

FIGURE 1.12 RSTP example 2

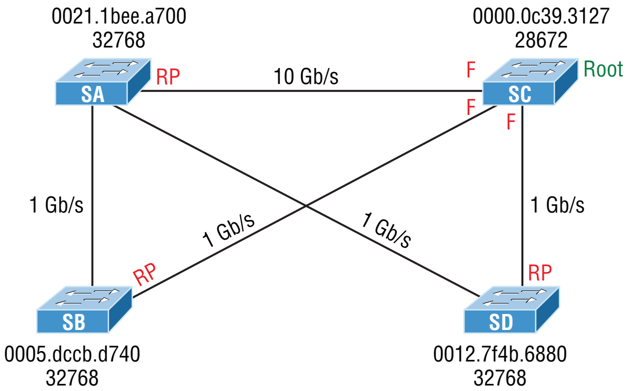

0021.1bee.a700
Priority: 32768
SA
0000.0c39.3127
Priority: 28672
SC
10 Gb/s
1 Gb/s
1 Gb/s
1 Gb/s
1 Gb/s
SB
0005.dccb.d740
Priority: 32768
SD
0012.7f4b.6880
Priority: 32768

So which bridge is our root bridge? Checking priorities first tells us that SC is the root bridge, which means all ports on SC are designated forwarding ports. Now we need to find our root ports.

We can quickly see that SA has a 10-gigabit port to SC, so that would be a port cost of 2, and it would be our root port. SD has a direct Gigabit Ethernet port to SC, so that would be the root port for SD with a port cost of 4. SB's best path would also be the direct Gigabit Ethernet port to SC with a port cost of 4.

Now that we've determined our root bridge and found the three root ports we need, we've got to find our designated ports next. Whatever is left over simply goes into the discarding role. Let's take a look at Figure 1.13 and see what we have.

FIGURE 1.13 RSTP example 2, answer 1


0021.1bee.a700
32768
SA
RP
10 Gb/s
F
0000.0c39.3127
28672
SC
Root
F
F
1 Gb/s
1 Gb/s
1 Gb/s
1 Gb/s
RP
RP
SB
0005.dccb.d740
32768
SD
0012.7f4b.6880
32768

All right, it looks like there are two links to choose between to find one designated port per segment. Let's start with the link between SA and SD. Which one has the best bridge ID? They're both running the same default priority, so by looking at the MAC address, we can see that SD has the better bridge ID (lower), so the SA port toward SD will go into a discarding role, or will it? The SD port will go into discarding mode, because the link from SA to the root has the lowest accumulated path costs to the root bridge, and that is used before the bridge ID in this circumstance. It makes sense to let the bridge with the fastest path to the root bridge be a designated forwarding port. Let's talk about this a little more in depth.

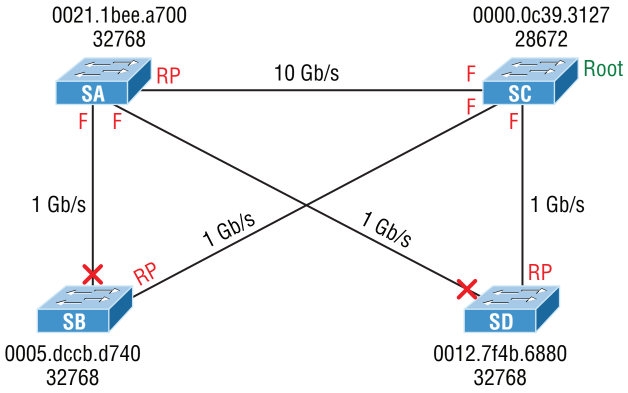
As you know, once your root bridge and root ports have been chosen, you're left with finding your designated ports. Anything left over goes into discarding role. But how are the designated ports chosen? Is it just bridge ID? Here are the rules:

1. To choose the switch that will forward on the segment, we select the switch with the lowest accumulated path cost to the root bridge. We want the fast path to the root bridge.
2. If there is a tie on the accumulated path cost from both switches to the root bridge, then we'll use bridge ID, which was what we used in our previous example (but not with this latest RSTP example; not with a 10-Gigabit Ethernet link to the root bridge available!).
3. Port priorities can be set manually if we want a specific port chosen. The default priority is 32, but we can lower that if needed.
4. If there are two links between switches, and the bridge ID and priority are tied, the port with the lowest number will be chosen—for example, Fa0/1 would be chosen over Fa0/2.

Let's take a look at our answer now, but before we do, can you find the forwarding port between SA and SB? Take a look at Figure 1.14 for the answer.

Again, to get the right answer to this question we're going to let the switch on the network segment with the lowest accumulated path cost to the root bridge forward on that segment. This is definitely SA, meaning the SB port goes into discarding role—not so hard at all!

FIGURE 1.14 RSTP example 2, answer 2



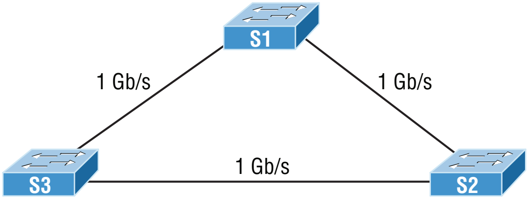
Visit ccna .gg/ch1/a for a companion MicroNugget from CBT Nuggets.

Modifying and Verifying the Bridge ID

To verify spanning tree on a Cisco switch, just use the command `show spanning-tree`. From its output, we can determine our root bridge, priorities, root ports, and designated and blocking/discarding ports.

Let's use the same simple three-switch network we used earlier as the base to play around with the configuration of STP. Figure 1.15 shows the network we'll work with in this section.

FIGURE 1.15 Our simple three-switch network



Let's start by taking a look at the output from S1:

```
S1#sh spanning-tree vlan 1
VLAN0001
  Spanning tree enabled protocol ieee
  Root ID    Priority    32769
             Address     0001.42A7.A603
             This bridge is the root
             Hello Time  2 sec  Max Age 20 sec  Forward Delay 15 sec


  Bridge ID  Priority    32769  (priority 32768 sys-id-ext 1)
             Address     0001.42A7.A603 him
             Hello Time  2 sec  Max Age 20 sec  Forward Delay 15 sec
             Aging Time  20


Interface        Role Sts Cost      Prio.Nbr Type
---------------- ---- --- --------- -------- --------------------------------
Gi1/1            Desg FWD 4         128.25   P2p
Gi1/2            Desg FWD 4         128.26   P2p
```

First, we can see that we're running the IEEE 802.1d STP version by default, and don't forget that this is really 802.1d PVST+! Looking at the output, we can see that

S1 is the root bridge for VLAN 1. When you use this command, the top information is about the root bridge, and the Bridge ID output refers to the bridge you're looking at. In this example, they are one and the same. Notice the `sys-id-ext 1 (for VLAN 1)`. This is the 12-bit PVST+ field that is placed into the BPDU so it can carry multiple-VLAN information. You add the priority and `sys-id-ext` to come up with the true priority for the VLAN. We can also see from the output that both Gigabit Ethernet interfaces are designated forwarding ports. You will not see a blocked/discarding port on a root bridge. Now let's take a look at S3's output:

```
S3#sh spanning-tree
VLAN0001
  Spanning tree enabled protocol ieee
  Root ID    Priority    32769
             Address     0001.42A7.A603
             Cost        4
             Port        26(GigabitEthernet1/2)
             Hello Time  2 sec  Max Age 20 sec  Forward Delay 15 sec


  Bridge ID  Priority    32769  (priority 32768 sys-id-ext 1)
             Address     000A.41D5.7937
             Hello Time  2 sec  Max Age 20 sec  Forward Delay 15 sec
             Aging Time  20


Interface        Role Sts Cost      Prio.Nbr Type
---------------- ---- --- --------- -------- --------------------------------
Gi1/1            Desg FWD 4         128.25   P2p
Gi1/2            Root FWD 4         128.26   P2p
```

Looking at the Root ID, it's easy to see that S3 isn't the root bridge, but the output tells us it's a cost of 4 to get to the root bridge and also that it's located out port 26 of the switch (Gi1/2). This tells us that the root bridge is one Gigabit Ethernet link away, which we already know is S1, but we can confirm this with the `show cdp neighbors` command:

```
Switch#sh cdp nei
Capability Codes: R - Router, T - Trans Bridge, B - Source Route Bridge
                  S - Switch, H - Host, I - IGMP, r - Repeater, P - Phone
Device ID    Local Intrfce   Holdtme    Capability   Platform    Port ID
S3           Gig 1/1          135            S       2960        Gig 1/1
S1           Gig 1/2          135            S       2960        Gig 1/1
```

That's how simple it is to find your root bridge if you don't have the nice figure as we do. Use the `show spanning-tree` command, find your root port, and then use the `show cdp neighbors` command. Let's see what S2's output has to tell us now:

```
S2#sh spanning-tree
VLAN0001
  Spanning tree enabled protocol ieee
  Root ID    Priority    32769
             Address     0001.42A7.A603
             Cost        4
             Port        26(GigabitEthernet1/2)
             Hello Time  2 sec  Max Age 20 sec  Forward Delay 15 sec


  Bridge ID  Priority    32769  (priority 32768 sys-id-ext 1)
             Address     0030.F222.2794
             Hello Time  2 sec  Max Age 20 sec  Forward Delay 15 sec
             Aging Time  20


Interface        Role Sts Cost      Prio.Nbr Type
---------------- ---- --- --------- -------- --------------------------------
Gi1/1            Altn BLK 4         128.25   P2p
Gi1/2            Root FWD 4         128.26   P2p
```

We're certainly not looking at a root bridge since we're seeing a blocked port, which is S2's connection to S3!

Let's have some fun by making S2 the root bridge for VLAN 2 and for VLAN 3. Here's how easy that is to do:

```
S2#sh spanning-tree vlan 2
VLAN0002
  Spanning tree enabled protocol ieee
  Root ID    Priority    32770
             Address     0001.42A7.A603
             Cost        4
             Port        26(GigabitEthernet1/2)
             Hello Time  2 sec  Max Age 20 sec  Forward Delay 15 sec


  Bridge ID  Priority    32770  (priority 32768 sys-id-ext 2)
             Address     0030.F222.2794
             Hello Time  2 sec  Max Age 20 sec  Forward Delay 15 sec
             Aging Time  20
```

```
Interface        Role Sts Cost      Prio.Nbr Type
---------------- ---- --- --------- -------- --------------------------------
Gi1/1            Altn BLK 4         128.25   P2p
Gi1/2            Root FWD 4         128.26   P2p
```

We can see that the root bridge cost is 4, meaning that the root bridge is one-gigabit link away. One more key factor I want to talk about before making S2 the root bridge for VLANs 2 and 3, is the `sys-id-ext`, which shows up as 2 in this output because this output is for VLAN 2. This `sys-id-ext` is added to the bridge priority, which in this case in 32768 + 2, which makes the priority 32770. Okay, now that you understand what that output is telling us, let's make S2 the root bridge:

```
S2(config)#spanning-tree vlan 2 ?
  priority  Set the bridge priority for the spanning tree
  root      Configure switch as root
  <cr>
S2(config)#spanning-tree vlan 2 priority ?
  <0-61440>  bridge priority in increments of 4096
S2(config)#spanning-tree vlan 2 priority 16384
```

You can set the priority to any value from 0 through 61440 in increments of 4096. Setting it to zero (0) means that the switch will always be a root as long as it has a lower MAC address than another switch that also has its bridge ID set to 0. If you want to set a switch to be the root bridge for every VLAN in your network, then you have to change the priority for each VLAN, with 0 being the lowest priority you can use. But trust me—it's never a good idea to set all switches to a priority of 0!

Furthermore, you don't actually need to change priorities because there is yet another way to configure the root bridge. Take a look:

```
S2(config)#spanning-tree vlan 3 root ?
  primary    Configure this switch as primary root for this spanning tree
  secondary  Configure switch as secondary root
S2(config)#spanning-tree vlan 3 root primary
```

Notice that you can set a bridge to either primary or secondary—very cool! Let's check to see if S2 is actually the root bridge for VLANs 2 and 3 now:

```
S2#sh spanning-tree vlan 2
VLAN0002
  Spanning tree enabled protocol ieee
  Root ID    Priority    16386
             Address     0030.F222.2794
             This bridge is the root
             Hello Time  2 sec  Max Age 20 sec  Forward Delay 15 sec
  Bridge ID  Priority    16386  (priority 16384 sys-id-ext 2)
             Address     0030.F222.2794
             Hello Time  2 sec  Max Age 20 sec  Forward Delay 15 sec
             Aging Time  20

Interface        Role Sts Cost      Prio.Nbr Type
---------------- ---- --- --------- -------- --------------------------------
Gi1/1            Desg FWD 4         128.25   P2p
Gi1/2            Desg FWD 4         128.26   P2p
```

Nice—S2 is the root bridge for VLAN 2, with a priority of 16386 (16384 + 2). Let's take a look to see the root bridge for VLAN 3. I'll use a different command for that this time. Check it out:

```
S2#sh spanning-tree summary
Switch is in pvst mode
Root bridge for: VLAN0002 VLAN0003
Extended system ID           is enabled
Portfast Default             is disabled
PortFast BPDU Guard Default  is disabled
Portfast BPDU Filter Default is disabled
Loopguard Default            is disabled
EtherChannel misconfig guard is disabled
UplinkFast                   is disabled
BackboneFast                 is disabled
Configured Pathcost method used is short

Name                   Blocking Listening Learning Forwarding STP Active
---------------------- -------- --------- -------- ---------- ----------
VLAN0001                     1         0        0          1          2
VLAN0002                     0         0        0          2          2
VLAN0003                     0         0        0          2          2


---------------------- -------- --------- -------- ---------- ----------
3 vlans                      1         0        0          5          6
```

The preceding output tells us that S2 is the root for the two VLANs, but we can see we have a blocked port for VLAN 1 on S2, so it's not the root bridge for VLAN 1. This is because there's another bridge with a better bridge ID for VLAN 1 than S2's.

One last burning question: How do you enable RSTP on a Cisco switch? Well, doing that is actually the easiest part of this chapter! Take a look:

```
S2(config)#spanning-tree mode rapid-pvst
```

Is that really all there is to it? Yes, because it's a global command, not per VLAN. Let's verify we're running RSTP now:

```
S2#sh spanning-tree
VLAN0001
  Spanning tree enabled protocol rstp
  Root ID    Priority    32769
             Address     0001.42A7.A603
             Cost        4
             Port        26(GigabitEthernet1/2)
             Hello Time  2 sec  Max Age 20 sec  Forward Delay 15 sec
[output cut
S2#sh spanning-tree summary
Switch is in rapid-pvst mode
Root bridge for: VLAN0002 VLAN0003
```

Looks like we're set! We're running RSTP, S1 is our root bridge for VLAN 1, and S2 is the root bridge for VLANs 2 and 3. I know this doesn't seem hard, and it really isn't, but you still need to practice what we've covered so far in this chapter to really get your skills solid!

Spanning-tree Failure Consequences

Clearly, there will be consequences when a routing protocol fails on a single router, but mainly, you'll just lose connectivity to the networks directly connected to that router and it usually does not affect the rest of your network. This definitely makes it easier to troubleshoot and fix the issue!

There are two failure types with STP. One of them causes the same type of issue I mentioned with a routing protocol, when certain ports have been placed in a blocking state they should be forwarding on a network segment instead. This situation makes the network segment unusable, but the rest of the network will still be working. But what happens when blocked ports are placed into forwarding state when they should be blocking? Let's work through this second failure issue now, using the same layout we used in the last section. Let's start with Figure 1.16 and then find out what happens when STP fails. Squeamish readers be warned—this isn't pretty!

Looking at Figure 1.16, what do you think will happen if SD transitions its blocked port to the forwarding state? Clearly, the consequences to the entire network will be pretty devastating! Frames that already had a destination address recorded in the MAC address table of the switches are forwarded to the port they're associated with; however, any broadcast, multicast, and unicasts not in the CAM are now in an endless loop. Figure 1.17 shows us the carnage—when you see all the lights on each port blinking super-fast amber/green, this means serious errors are occurring, and lots of them!

FIGURE 1.16 STP stopping loops

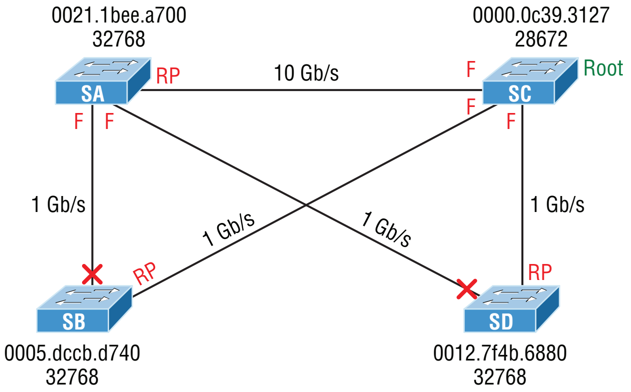
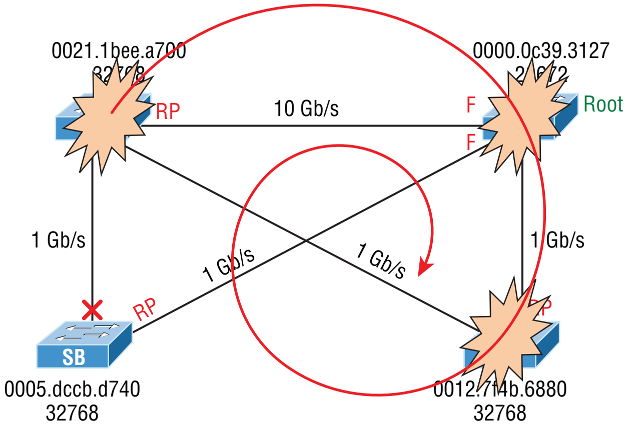

0021.1bee.a700
32768
0000.0c39.3127
28672
Root
SA
SC
RP
10 Gb/s
F
F
F
F
F
1 Gb/s
1 Gb/s
1 Gb/s
1 Gb/s
RP
RP
SB
SD
0005.dccb.d740
32768
0012.7f4b.6880
32768

FIGURE 1.17 STP failure


0021.1bee.a700
0000.0c39.3127
Root
RP
10 Gb/s
F
F
1 Gb/s
1 Gb/s
1 Gb/s
1 Gb/s
RP
SB
0005.dccb.d740
32768
0012.7f4b.6880
32768

As frames begin building up on the network, the bandwidth starts getting saturated. The CPU percentage goes way up on the switches until they'll just give up and stop working completely, and all this within a few seconds!

Here is a list of the problems that will occur in a failed STP network that you must be aware of and you must be able to find in your production network—and of course, you must know them to meet the exam objectives:

- The load on all links begins increasing and more and more frames enter the loop. Remember, this loop affects all the other links in the network because these frames are always flooded out all ports. This scenario is a little less dire if the loop occurs within a single VLAN. In that case, the snag will be isolated to ports only in that VLAN membership, plus all trunk links that carry information for that VLAN.

- If you have more than one loop, traffic will increase on the switches because all the circling frames actually get duplicated. Switches basically receive a frame, make a copy of it, and send it out all ports. And they do this over and over and over again with the same frame, as well as for any new ones!
- The MAC address table is now completely unstable. It no longer knows where any source MAC address hosts are actually located because the same source address comes in via multiple ports on the switch.
- With the overwhelmingly high load on the links and the CPUs, now possibly at 100% or close to that, the devices become unresponsive, making it impossible to troubleshoot—it's a terrible thing!

At this point your only option is to systematically remove every redundant link between switches until you can find the source of the problem. And don't freak because, eventually, your ravaged network will calm down and come back to life after STP converges. Your fried switches will regain consciousness, but the network will need some serious therapy, so you're not out of the woods yet!

Now is when you start troubleshooting to find out what caused the disaster in the first place. A good strategy is to place the redundant links back into your network one at a time and wait to see when a problem begins to occur. You could have a failing switch port, or even a dead switch. Once you've replaced all your redundant links, you need to carefully monitor the network and have a back-out plan to quickly isolate the problem if it reoccurs. You don't want to go through this again!

You're probably wondering how to prevent these STP problems from ever darkening your doorstep in the first place. Well, just hang on, because after the next section, I'll tell you all about EtherChannel, which can stop ports from being placed in the blocked/discarding state on redundant links to save the day! But before we add more links to our switches and then bundle them, let's talk about PortFast.

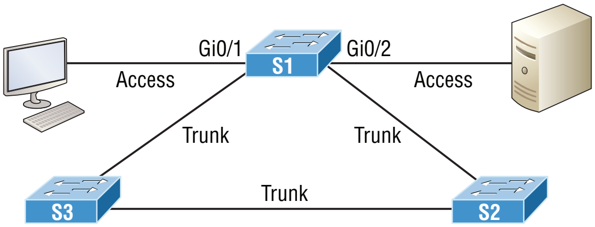
PortFast and BPDU Guard

If you have a server or other devices connected into your switch that you're totally sure won't create a switching loop if STP is disabled, you can use a Cisco proprietary extension to the 802.1d standard called PortFast on these ports. With this tool, the port won't spend the usual 50 seconds to come up into forwarding mode while STP is converging, which is what makes it so cool.

Since ports will transition from blocking to forwarding state immediately, PortFast can prevent our hosts from being potentially unable to receive a DHCP address due to STP's slow convergence. If the host's DHCP request times out, or if every time you plug a host in you're just tired of looking at the switch port being amber for almost a minute before it transitions to forwarding state and turns green, PortFast can really help you out!

Figure 1.18 illustrates a network with three switches, each with a trunk to each of the others and a host and server off the S1 switch.

FIGURE 1.18 PortFast



We can use PortFast on the ports on S1 to help them transition to the STP forwarding state immediately upon connecting to the switch.

Here are the commands, first from global config mode—they're pretty simple:

```
S1(config)#spanning-tree portfast ?
  bpdufilter  Enable portfast bdpu filter on this switch
  bpduguard   Enable portfast bpdu guard on this switch
  default     Enable portfast by default on all access ports
```

If you were to type `spanning-tree portfast default`, you would enable all nontrunking ports with PortFast. From interface mode, you can be more specific, which is the better way to go:

```
S1(config-if)#spanning-tree portfast ?
  disable  Disable portfast for this interface
  trunk    Enable portfast on the interface even in trunk mode
  <cr>
```

From interface mode you can actually configure PortFast on a trunk port, but you would do that only if the port connects to a server or router, not to another switch, so we won't use that here. So let's take a look at the message I get when I turn on PortFast on an interface Gi0/1:

```
S1#config t
S1#config)#int range gi0/1 - 2
S1(config-if)#spanning-tree portfast
%Warning: portfast should only be enabled on ports connected to a single
 host. Connecting hubs, concentrators, switches, bridges, etc... to this
 interface  when portfast is enabled, can cause temporary bridging loops.
 Use with CAUTION


%Portfast has been configured on GigabitEthernet0/1 but will only
 have effect when the interface is in a non-trunking mode.
Switch(config-if)#
```

PortFast is enabled on port Gi0/1 and Gi0/2, but notice that you get a pretty long message that's essentially telling you to be careful. This is because when using PortFast, you definitely don't want to create a network loop by plugging another switch or hub into a port that's also configured with PortFast! Why? Because if you let this happen, even though the network may still sort of work, data will pass super slowly, and worse, it could take you a really long time to find the source of the problem, making you very unpopular. So proceed with caution!

At this juncture, you would be happy to know that there are some safeguard commands to have handy when using PortFast just in case someone causes a loop in a port that's configured with PortFast enabled. Let's talk a really key safeguard command now.

BPDU Guard

If you turn on PortFast for a switch port, it's a really good idea to turn on BPDU Guard as well. In fact, it's such a great idea, I personally feel that it should be enabled by default whenever a port is configured with PortFast!

This is because if a switch port that has PortFast enabled receives a BPDU on that port, it will place the port into error disabled (shutdown) state, effectively preventing anyone from accidentally connecting another switch or hub port into a switch port configured with PortFast. Basically, you're preventing (guarding) your network from being severely crippled or even brought down. So let's configure our S1 interface, which is already configured with PortFast, with BPDU Guard now—it's easy!

Here's how to set it globally:

Visit `ccna.gg/ch1/b` for a companion MicroNugget from CBT Nuggets.

```
S1(config)# spanning-tree portfast bpduguard default
```

And specifically on an interface:

```
S1(config-if)#spanning-tree bpduguard enable
```

It's important to know that you would only configure this command on your access layer switches—switches where users are directly connected.

Real World Scenario

Hedging My Bets Created Bad Switch Ports During the Super Bowl

A junior admin called me frantically telling me all switch ports have just gone bad on the core switch, which was located at the data center where I was lead consultant for a data center upgrade. Now these things happen, but keep in mind that I just happened to be at a Super Bowl party having a great time watching my favorite team play in the "Big One" when I received this call! So I took a deep breath to refocus. I needed to find out some key information to determine just how bad the situation really was, and my client was in as big of a hurry as I was to get to a solution!

First I asked the junior admin exactly what he did. Of course, he said, “Nothing, I swear!” I figured that’s what he’d say, so I pressed him for more info and finally asked for stats on the switch. The admin told me that all the ports on the 10/100/1000 line card went amber at the same time—finally some information I could use! I confirmed that, as suspected, these ports trunked to uplink distribution switches. Okay, wow—this was not good!

At this point, though, I found it hard to believe that all 24 ports would suddenly go bad, but it’s possible, so I asked if he had a spare card to try. He told me that he had already put in the new card but the same thing was still happening. Well, okay—it’s not the card, or the ports, but maybe something happened with the other switches. I knew there were a lot of switches involved, so someone must have screwed something up to make this catastrophe happen! Or, maybe the fiber distribution closet went down somehow? If so, how? Was there a fire in the closet or something? Some serious internal shenanigans would be the only answer if that were the cause!

So remaining ever-patient (because, to quote Dr. House, “Patients lie”), I again had to ask the admin exactly what he did, and sure enough, he finally admitted that he tried to plug his personal laptop into the core switch so he could watch the Super Bowl, and quickly added, “…but that’s it, I didn’t do anything else!” I’ll skip over the fact that this guy was about to have the ugliest Monday ever, but something still didn’t make sense, and here’s why.

Knowing that the ports on that card would all connect to distribution switches, I configured the ports with PortFast so they wouldn’t have to transition through the STP process. And because I wanted to make sure no one plugged a switch into any of those ports, I enabled BPDU Guard on the entire line card.

But a host would not bring down those ports, so I asked him if he had plugged in the laptop directly or used something in between. He admitted that he had indeed used another switch because, turns out, there were lots of people from the office who wanted to plug into the core switch and watch the game too. Was he kidding me? The security policy wouldn’t allow connecting from their offices, so wouldn’t you think they’d consider the core even more off-limits? Some people!

But wait… This doesn’t explain all ports turning amber, because only the one he plugged into should be doing that. It took me a second, but I figured out what he did and finally got him to confess. When he plugged the switch in, the port turned amber so he thought it went bad. So what do think he did? Well, if at first you don’t succeed, try, try again, and that’s just what he did—he actually kept trying ports—all 24 of them to be exact! Now that’s what I call determined!

Sad to say, I got back to the party in time just to watch my team lose in the last few minutes! A dark day, indeed!

EtherChannel

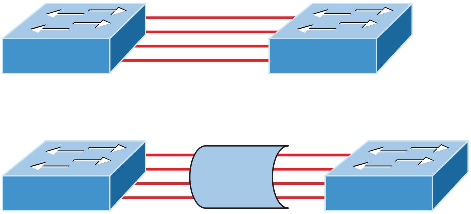
Know that almost all Ethernet networks today will typically have multiple links between switches because this kind of design provides redundancy and resiliency. On a physical design that includes multiple links between switches, STP will do its job and put a port or ports into blocking mode. In addition to that, routing protocols like OSPF and EIGRP could see all these redundant links as individual ones, depending on the configuration, which can mean an increase in routing overhead.

We can gain the benefits from multiple links between switches by using port channeling. EtherChannel is a port channel technology that was originally developed by Cisco as a switch-to-switch technique for grouping several Fast Ethernet or Gigabit Ethernet ports into one logical channel.

Also important to note is that once your port channel (EtherXhannel) is up and working, layer 2 STP and layer 3 routing protocols will treat those bundled links as a single one, which would stop STP from performing blocking. An additional nice result is that because the routing protocols now only see this as a single link, a single adjacency across the link can be formed—elegant!

Figure 1.19 shows how a network would look if we had four connections between-switches, before and after configuring port channels.

FIGURE 1.19 Before and after port channels



Okay—now as usual, there's the Cisco version and the IEEE version of port channel negotiation protocols to choose from—take your pick. Cisco's version is called Port Aggregation Protocol (PAgP), and the IEEE 802.3ad standard is called Link Aggregation Control Protocol (LACP). Both versions work equally well, but the way you configure each is slightly different. Keep in mind that both PAgP and LACP are negotiation protocols and that EtherChannel can actually be statically configured without PAgP or LACP. Still, it's better to use one of these protocols to help with compatibility issues as well as to manage link additions and failures between two switches.

Cisco EtherChannel allows us to bundle up to 8 ports active between switches. The links must have the same speed, duplex setting, and VLAN configuration—in other words you can't mix interface types and configurations into the same bundle.

There are a few differences in configuring PAgP and LACP, but first, let's go over some terms so you don't get confused:

Port channeling Refers to combining two-to-eight Fast Ethernet or two-Gigabit Ethernet ports together between two switches into one aggregated logical link to achieve more bandwidth and resiliency.

EtherChannel Cisco's proprietary term for port channeling.

PAgP This is a Cisco proprietary port channel negotiation protocol that aids in the automatic creation for EtherChannel links. All links in the bundle must match the same parameters (speed, duplex, VLAN info), and when PAgP identifies matched links, it groups the links into an EtherChannel. This is then added to STP as a single bridge port. At this point, PAgP's job is to send packets every 30 seconds to manage the link for consistency, any link additions, and failures.

LACP (802.3ad) This has the exact same purpose as PAgP, but is nonproprietary so it can work between multi-vendor networks.

`Channel-group` This is a command on Ethernet interfaces used to add the specified interface to a single EtherChannel. The number following this command is the port channel ID.

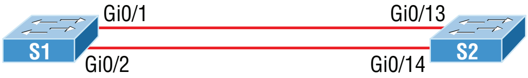
`Interface port-channel` Here's a command that creates the bundled interface. Ports can be added to this interface with the `channel-group` command. Keep in mind that the interface number must match the group number.

Okay, now let's see if you can make some sense out of all these terms by actually configuring something!

Configuring and Verifying Port Channels

Let's use Figure 1.20 for our simple example of how to configure port channels.

FIGURE 1.20 EtherChannel example



You can enable your `channel-group` for each channel by setting the channel mode for each interface to either `active` or `passive` if using LACP. When a port is configured in `passive` mode, it will respond to the LACP packets it receives, but it won't initiate an LACP negotiation. When a port is configured for `active` mode, the port initiates negotiations with other ports by sending LACP packets.

Let me show you a simple example of configuring port channels and then verifying them. First I'll go to global configuration mode and create a port channel interface, and then I'll add this port channel to the physical interfaces.

Remember, all parameters and configurations of the ports must be the same, so I'll start by trunking the interfaces before I configure EtherChannel, like this:

```
S1(config)#int range g0/1 - 2
S1(config-if-range)#switchport trunk encapsulation dot1q
S1(config-if-range)#switchport mode trunk
```

All ports in your bundles must be configured the same, so I'll configure both sides with the same trunking configuration. Now I can assign these ports to a bundle:

```
S1(config-if-range)#channel-group 1 mode ?
  active     Enable LACP unconditionally
  auto       Enable PAgP only if a PAgP device is detected
  desirable  Enable PAgP unconditionally
  on         Enable Etherchannel only
  passive    Enable LACP only if a LACP device is detected
S1(config-if-range)#channel-group 1 mode active
S1(config-if-range)#exit
```

To configure the IEEE nonproprietary LACP, I'll use the `active` or `passive` command; if I wanted to use Cisco's PAgP, I'd use the `auto` or `desirable` command. You can't mix and match these on either end of the bundle, and really, it doesn't matter which one you use in a pure Cisco environment, as long as you configure them the same on both ends (setting the mode to on would be statically configuring your EtherChannel bundle). At this point in the configuration, I'd have to set the mode to `active` on the S2 interfaces if I wanted the bundle to come up with LACP because again, all parameters must be the same on both ends of the link. Let's create our port channel interface now with the `interface port-channel` command:

```
S1(config)#int port-channel 1
S1(config-if)#switchport trunk encapsulation dot1q
S1(config-if)#switchport mode trunk
S1(config-if)#switchport trunk allowed vlan 1,2,3
```

Notice that I set the same trunking method under the port channel interface as I did the physical interfaces, as well as VLAN information too.

Time to configure the interfaces, channel groups, and port channel interface on the S2 switch:

```
S2(config)#int range g0/13 - 14
S2(config-if-range)#switchport trunk encapsulation dot1q
S2(config-if-range)#switchport mode trunk
S2(config-if-range)#channel-group 1 mode active
S2(config-if-range)#exit
S2(config)#int port-channel 1
S2(config-if)#switchport trunk encapsulation dot1q
```

```
S2(config-if)#switchport mode trunk
S2(config-if)#switchport trunk allowed vlan 1,2,3
```

On each switch, I configured the ports I wanted to bundle with the same configuration, then created the port channel. After that, I added the ports into the port channel with the `channel-group` command.

Let's verify our EtherChannel with a few commands. We'll start with the `show etherchannel port-channel` command to see information about a specific port channel interface:

```
S2#sh etherchannel port-channel
                Channel-group listing:
                ----------------------


Group: 1
----------
                Port-channels in the group:
                ---------------------------


Port-channel: Po1    (Primary Aggregator)
------------


Age of the Port-channel   = 00d:00h:46m:49s
Logical slot/port   = 2/1       Number of ports = 2
GC                  = 0x00000000      HotStandBy port = null
Port state          = Port-channel
Protocol            =   LACP
Port Security       = Disabled


Ports in the Port-channel:


Index   Load   Port     EC state        No of bits
------+------+------+------------------+-----------
  0     00     Gig0/2   Active             0
  0     00     Gig0/1   Active             0
Time since last port bundled:    00d:00h:46m:47s    Gig0/1
S2#
```

Notice that we have one group and that we're running the IEEE LACP version of port channeling. We're in `Active` mode, and that `Port-channel: Po1` interface has two physical interfaces. The heading `Load` is not the load over the interfaces, it's a hexadecimal value that decides which interface will be chosen to specify the flow of traffic.

The `show etherchannel summary` command displays one line of information per port channel:

```
S2#sh etherchannel summary
Flags:  D - down        P - in port-channel
        I - stand-alone s - suspended
        H - Hot-standby (LACP only)
        R - Layer3      S - Layer2
        U - in use      f - failed to allocate aggregator
        u - unsuitable for bundling
        w - waiting to be aggregated
        d - default port


Number of channel-groups in use: 1
Number of aggregators:           1


Group  Port-channel  Protocol    Ports
------+-------------+-----------+-----------------------------------------------


1      Po1(SU)           LACP   Gig0/1(P) Gig0/2(P)
```

This command shows that we have one group, that we're running LACP, and Gig0/1 and Gig0/2 or (P), which means these ports are `in port-channel` mode. This command isn't really all that helpful unless you have multiple channel groups, but it does tell us our group is working well!

Summary

This chapter was all about switching technologies, with a particular focus on the Spanning Tree Protocol (STP) and its evolution to newer versions like RSTP and then Cisco's PVST+.

You learned about the problems that can occur if you have multiple links between bridges (switches) and the solutions attained with STP.

I also talked about and demonstrated issues that can occur if you have multiple links between bridges (switches), plus how to solve these problems by using the Spanning Tree Protocol (STP).

I covered a detailed configuration of Cisco's Catalyst switches, including verifying the configuration, setting the Cisco STP extensions, and changing the root bridge by setting a bridge priority.

Finally, we discussed, configured, and verified the EtherChannel technology that helps us bundle multiple links between switches.

Exam Essentials

Understand the main purpose of the Spanning Tree Protocol in a switched LAN. The main purpose of STP is to prevent switching loops in a network with redundant switched paths.

Remember the states of STP. The purpose of the blocking state is to prevent the use of looped paths. A port in listening state prepares to forward data frames without populating the MAC address table. A port in learning state populates the MAC address table but doesn't forward data frames. A port in forwarding state sends and receives all data frames on the bridged port. Also, a port in the disabled state is virtually nonoperational.

Remember the command `show spanning-tree`. You must be familiar with the command `show spanning-tree` and how to determine the root bridge of each VLAN. Also, you can use the `show spanning-tree summary` command to help you get a quick glimpse of your STP network and root bridges.

Understand what PortFast and BPDU Guard provides. PortFast allows a port to transition to the forwarding state immediately upon a connection. Because you don't want other switches connecting to this port, BPDU Guard will shut down a PortFast port if is receives a BPDU.

Understand what EtherChannel is and how to configure it. EtherChannel allows you to bundle links to get more bandwidth, instead of allowing STP to shut down redundant ports. You can configure Cisco's PAgP or the IEEE version, LACP, by creating a port channel interface and assigning the port channel group number to the interfaces you are bundling.

Written Lab 1

The answers to this lab can be found in Appendix A, “Answers to Written Labs.”

Write the answers to the following questions:

1. Which of the following is Cisco proprietary: LACP or PAgP?
2. What command will show you the STP root bridge for a VLAN?
3. What standard is RSTP PVST+ based on?
4. Which protocol is used in a layer 2 network to maintain a loop-free network?
5. Which proprietary Cisco STP extension would put a switch port into error-mode disabled mode if a BPDU is received on this port?
6. You want to configure a switch port to not transition through the STP port states but to go immediately to forwarding mode. What command will you use on a per-port basis?
7. What command will you use to see information about a specific port channel interface?
8. What command can you use to set a switch so that it will be the root bridge for VLAN 3 over any other switch?
9. You need to find the VLANs for which your switch is the root bridge. What two commands can you use?
10. What are the two modes you can set with LACP?

Hands-on Labs

In this section, you will use configure and verify STP, as well as configure PortFast and BPDU Guard, and finally bundle links together with EtherChannel.

Note that the labs in this chapter were written to be used with real equipment using 2960 switches, however, you can use the free LammleSim IOS version simulator or Cisco’s Packet Tracer to run through these labs.

The labs in this chapter are as follows:

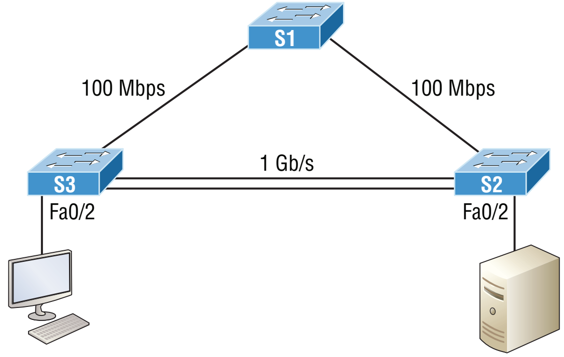
Lab 1.1: Verifying STP and Finding Your Root Bridge

Lab 1.2: Configuring and Verifying Your Root Bridge

Lab 1.3: Configuring PortFast and BPDU Guard

Lab 1.4: Configuring and Verifying EtherChannel

We'll use the following illustration for all four labs:



Hands-on Lab 1.1: Verifying STP and Finding Your Root Bridge

This lab will assume that you have added VLANs 2 and 3 to each of your switches and all of your links are trunked.

1. From one of your switches, use the `show spanning-tree vlan 2` command. Verify the output.

```
S3#sh spanning-tree vlan 2
VLAN0002
  Spanning tree enabled protocol ieee
  Root ID    Priority    32770
             Address     0001.C9A5.8748
             Cost        19
             Port        1(FastEthernet0/1)
             Hello Time  2 sec  Max Age 20 sec  Forward Delay 15 sec


  Bridge ID  Priority    32770  (priority 32768 sys-id-ext 2)
             Address     0004.9A04.ED97
             Hello Time  2 sec  Max Age 20 sec  Forward Delay 15 sec
             Aging Time  20


Interface        Role Sts Cost      Prio.Nbr Type
---------------- ---- --- --------- -------- --------------------------------
Fa0/1            Root FWD 19        128.1    P2p
Fa0/2            Desg FWD 19        128.2    P2p
```

```
Gi1/1            Altn BLK 4          128.25   P2p
Gi1/2            Altn BLK 4          128.26   P2p
```

Notice that S3 is not the root bridge, so to find your root bridge, just follow the root port and see what bridge is connected to that port. Port Fa0/1 is the root port with a cost of 19, which means the switch that is off the Fa0/1 port is the root port connecting to the root bridge because it is a cost of 19, meaning one Fast Ethernet link away.

2. Find the bridge that is off of Fa0/1, which will be our root.

```
S3#sh cdp neighbors
Capability Codes: R - Router, T - Trans Bridge, B - Source Route Bridge
                  S - Switch, H - Host, I - IGMP, r - Repeater, P - Phone
Device ID    Local Intrfce   Holdtme    Capability   Platform    Port ID
S1           Fas 0/1          158           S        2960        Fas 0/1
S2           Gig 1/1          151           S        2960        Gig 1/1
S2           Gig 1/2          151           S        2960        Gig 1/2
S3#
```

Notice that S1 is connected to the local interface Fa0/1, so let's go to S1 and verify our root bridge.

3. Verify the root bridge for each of the three VLANs. From S1 use the `show spanning-tree summary` command.

```
S1#sh spanning-tree summary
Switch is in pvst mode
Root bridge for: default VLAN0002 VLAN0003
Extended system ID           is enabled
Portfast Default             is disabled
PortFast BPDU Guard Default  is disabled
Portfast BPDU Filter Default is disabled
Loopguard Default            is disabled
EtherChannel misconfig guard is disabled
UplinkFast                   is disabled
BackboneFast                 is disabled
Configured Pathcost method used is short

Name                   Blocking Listening Learning Forwarding STP Active
---------------------- -------- --------- -------- ---------- ----------
VLAN0001                     0         0        0          2          2
VLAN0002                     0         0        0          2          2
VLAN0003                     0         0        0          2          2
```

```
---------------------- -------- --------- -------- ---------- ----------
3 vlans                      0         0        0          6          6


S1#
```

Notice that S1 is the root bridge for all three VLANs.

4. Make note of all your root bridges, for all three VLANs, if you have more than one root bridge.

Hands-on Lab 1.2: Configuring and Verifying Your Root Bridge

This lab will assume you have performed Lab 1 and know who your root bridge is for each VLAN.

1. Go to one of your non-root bridges and verify the bridge ID with the `show spanning-tree vlan` command.

```
S3#sh spanning-tree vlan 1
VLAN0001
  Spanning tree enabled protocol ieee
  Root ID    Priority    32769
             Address     0001.C9A5.8748
             Cost        19
             Port        1(FastEthernet0/1)
             Hello Time  2 sec  Max Age 20 sec  Forward Delay 15 sec


  Bridge ID  Priority    32769  (priority 32768 sys-id-ext 1)
             Address     0004.9A04.ED97
             Hello Time  2 sec  Max Age 20 sec  Forward Delay 15 sec
             Aging Time  20


Interface        Role Sts Cost      Prio.Nbr Type
---------------- ---- --- --------- -------- --------------------------------
Fa0/1            Root FWD 19        128.1    P2p
Fa0/2            Desg FWD 19        128.2    P2p
Gi1/1            Altn BLK 4         128.25   P2p
Gi1/2            Altn BLK 4         128.26   P2p
```

Notice that this bridge is not the root bridge for VLAN 1, and the root port is Fa0/1 with a cost of 19, which means the root bridge is directly connected one Fast Ethernet link away.

2. Make one of your non-root bridges the root bridge for VLAN 1. Use priority 16,384, which is lower than the 32,768 of the current root.

```
S3(config)#spanning-tree vlan 1 priority ?
  <0-61440>  bridge priority in increments of 4096
S3(config)#spanning-tree vlan 1 priority 16384
```

3. Verify the root bridge for VLAN 1.

```
S3#sh spanning-tree vlan 1
VLAN0001
  Spanning tree enabled protocol ieee
  Root ID    Priority    16385
             Address     0004.9A04.ED97
             This bridge is the root
             Hello Time  2 sec  Max Age 20 sec  Forward Delay 15 sec


  Bridge ID  Priority    16385  (priority 16384 sys-id-ext 1)
             Address     0004.9A04.ED97
             Hello Time  2 sec  Max Age 20 sec  Forward Delay 15 sec
             Aging Time  20


Interface        Role Sts Cost      Prio.Nbr Type
---------------- ---- --- --------- -------- --------------------------------
Fa0/1            Desg FWD 19        128.1    P2p
Fa0/2            Desg FWD 19        128.2    P2p
Gi1/1            Desg FWD 4         128.25   P2p
Gi1/2            Desg FWD 4         128.26   P2p
```

Notice that this bridge is indeed the root and all ports are in Desg FWD mode.

Hands-on Lab 1.3: Configuring PortFast and BPDU Guard

This lab will have you configure ports on switches S3 and S2 to allow the PC and server to automatically go into forward mode when they connect into the port.

1. Connect to your switch that has a host connected and enable PortFast for the interface.

```
S3#config t
S3(config)#int fa0/2
S3(config-if)#spanning-tree portfast
%Warning: portfast should only be enabled on ports connected to a single
host. Connecting hubs, concentrators, switches, bridges, etc... to this
```

```
interface  when portfast is enabled, can cause temporary bridging loops.
Use with CAUTION


%Portfast has been configured on FastEthernet0/2 but will only
have effect when the interface is in a non-trunking mode.
```

2. Verify that that port will be shut down if another switch plugs into this port.

```
S3(config-if)#spanning-tree bpduguard enable
```

3. Verify your configuration with the `show running-config` command.

```
!
interface FastEthernet0/2
 switchport mode trunk
 spanning-tree portfast
 spanning-tree bpduguard enable
!
```

Hands-on Lab 1.4: Configuring and Verifying EtherChannel

This lab will have you configure the Cisco EtherChannel PAgP version on the switches used in this lab. Because I have preconfigured the switches, I have set up the trunks on all inter-switch ports. We'll use the Gigabit Ethernet ports between switches S3 and S2.

1. Configure the S3 switch with EtherChannel by creating a port channel interface.

```
S3#config t
S3(config)#inter port-channel 1
```

2. Configure the ports to be in the bundle with the `channel-group` command.

```
S3(config-if)#int range g1/1 - 2
S3(config-if-range)#channel-group 1 mode ?
  active     Enable LACP unconditionally
  auto       Enable PAgP only if a PAgP device is detected
  desirable  Enable PAgP unconditionally
  on         Enable Etherchannel only
  passive    Enable LACP only if a LACP device is detected
S3(config-if-range)#channel-group 1 mode desirable
```

I chose the PAgP desirable mode for the S3 switch.

3. Configure the S2 switch with EtherChannel, using the same parameters as S3.

```
S2#config t
S2(config)#interface port-channel 1
S2(config-if)#int rang g1/1 - 2
S2(config-if-range)#channel-group 1 mode desirable
%LINK-5-CHANGED: Interface Port-channel 1, changed state to up


%LINEPROTO-5-UPDOWN: Line protocol on Interface Port-channel 1, changed state
to up
```

Pretty simple, really. Just a couple of commands.

4. Verify with the `show etherchannel port-channel` command.

```
S3#sh etherchannel port-channel
                Channel-group listing:
                ----------------------


Group: 1
----------
                Port-channels in the group:
                ---------------------------


Port-channel: Po1
------------


Age of the Port-channel   = 00d:00h:06m:43s
Logical slot/port   = 2/1       Number of ports = 2
GC                  = 0x00000000      HotStandBy port = null
Port state          = Port-channel
Protocol            =   PAGP
Port Security       = Disabled


Ports in the Port-channel:


Index   Load   Port     EC state        No of bits
------+------+------+------------------+-----------
  0     00     Gig1/1   Desirable-Sl       0
  0     00     Gig1/2   Desirable-Sl       0
Time since last port bundled:    00d:00h:01m:30s    Gig1/2
```

5. Verify with the `show etherchannel summary` command.

```
S3#sh etherchannel summary
Flags:  D - down        P - in port-channel
        I - stand-alone s - suspended
        H - Hot-standby (LACP only)
        R - Layer3      S - Layer2
        U - in use      f - failed to allocate aggregator
        u - unsuitable for bundling
        w - waiting to be aggregated
        d - default port


Number of channel-groups in use: 1
Number of aggregators:           1


Group  Port-channel  Protocol    Ports
------+-------------+-----------+----------------------------------


1      Po1(SU)           PAgP   Gig1/1(P) Gig1/2(P)
S3#
```

Review Questions

The following questions are designed to test your understanding of this chapter's material. For more information on how to get additional questions, please see this book's introduction.

The answers to these questions can be found in Appendix B, "Answers to Review Questions."

1. You receive the following output from a switch:

```
S2#sh spanning-tree
VLAN0001
  Spanning tree enabled protocol rstp
  Root ID    Priority    32769
             Address     0001.42A7.A603
             Cost        4
             Port        26(GigabitEthernet1/2)
             Hello Time  2 sec  Max Age 20 sec  Forward Delay 15 sec
[output cut]
```

Which are true regarding this switch? (Choose two.)

- **A.** The switch is a root bridge.
- **B.** The switch is a non-root bridge.
- **C.** The root bridge is four switches away.
- **D.** The switch is running 802.1w.
- **E.** The switch is running STP PVST+.

2. If you want to effectively disable STP on a port connected to a server, which command would you use?

- **A.** `disable spanning-tree`
- **B.** `spanning-tree off`
- **C.** `spanning-tree security`
- **D.** `spanning-tree portfast`

3. Which of the following would you use to find the VLANs for which your switch is the root bridge? (Choose two.)

- **A.** `show spanning-tree`
- **B.** `show root all`
- **C.** `show spanning-tree port root VLAN`
- **D.** `show spanning-tree summary`

4. You want to run the new 802.1w on your switches. Which of the following would enable this protocol?

- **A.** `Switch(config)#spanning-tree mode rapid-pvst`
- **B.** `Switch#spanning-tree mode rapid-pvst`
- **C.** `Switch(config)#spanning-tree mode 802.1w`
- **D.** `Switch#spanning-tree mode 802.1w`

5. Which of the following is a layer 2 protocol used to maintain a loop-free network?

- **A.** VTP
- **B.** STP
- **C.** RIP
- **D.** CDP

6. Which statement describes a spanning-tree network that has converged?

- **A.** All switch and bridge ports are in the forwarding state.
- **B.** All switch and bridge ports are assigned as either root or designated ports.
- **C.** All switch and bridge ports are in either the forwarding or blocking state.
- **D.** All switch and bridge ports are either blocking or looping.

7. Which of the following modes enable LACP EtherChannel? (Choose two.)

- **A.** On
- **B.** Prevent
- **C.** Passive
- **D.** Auto
- **E.** Active
- **F.** Desirable

8. Which of the following are true regarding RSTP? (Choose three.)

A. RSTP speeds the recalculation of the spanning tree when the layer 2 network topology changes.

B. RSTP is an IEEE standard that redefines STP port roles, states, and BPDUs.

C. RSTP is extremely proactive and very quick, and therefore it absolutely needs the 802.1 delay timers.

D. RSTP (802.1w) supersedes 802.1d while remaining proprietary.

E. All of the 802.1d terminology and most parameters have been changed.

F. 802.1w is capable of reverting to 802.1d to interoperate with traditional switches on a per-port basis.

9. What does BPDU Guard perform?

A. Makes sure the port is receiving BPDUs from the correct upstream switch.

B. Makes sure the port is not receiving BPDUs from the upstream switch, only the root.

C. If a BPDU is received on a BPDU Guard port, PortFast is used to shut down the port.

D. Shuts down a port if a BPDU is seen on that port.

10. How many bits is the sys-id-ext field in a BPDU?

A. 4

B. 8

C. 12

D. 16

11. There are four connections between two switches running RSTP PVST+ and you want to figure out how to achieve higher bandwidth without sacrificing the resiliency that RSTP provides. What can you configure between these two switches to achieve higher bandwidth than the default configuration is already providing?

A. Set PortFast and BPDU Guard, which provides faster convergence.

B. Configure unequal cost load balancing with RSTP PVST+.

C. Place all four links into the same EtherChannel bundle.

D. Configure PPP and use multilink.

12. In which circumstance are multiple copies of the same unicast frame likely to be transmitted in a switched LAN?

A. During high-traffic periods

B. After broken links are reestablished

C. When upper-layer protocols require high reliability

D. In an improperly implemented redundant topology

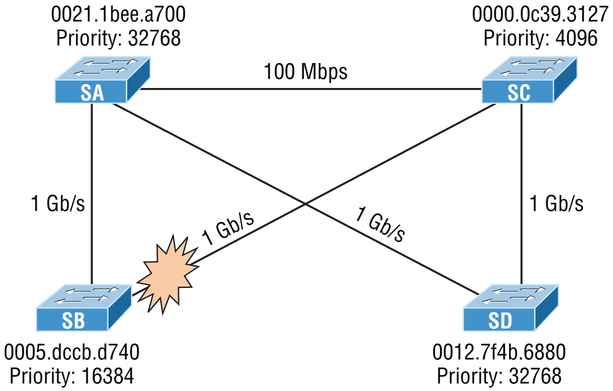
13. You want to configure LACP. Which do you need to make sure are configured exactly the same on all switch interfaces you are using? (Choose three.)

- **A.** Virtual MAC address
- **B.** Port speeds
- **C.** Duplex
- **D.** PortFast enabled
- **E.** VLAN information

14. Which of the following modes enable PAgP EtherChannel? (Choose two.)

- **A.** On
- **B.** Prevent
- **C.** Passive
- **D.** Auto
- **E.** Active
- **F.** Desirable

15. For this question, refer to the following illustration. SB's RP to the root bridge has failed.


0021.1bee.a700
Priority: 32768
0000.0c39.3127
Priority: 4096
100 Mbps
SA
SC
1 Gb/s
1 Gb/s
1 Gb/s
1 Gb/s
SB
SD
0005.dccb.d740
Priority: 16384
0012.7f4b.6880
Priority: 32768

What is the new cost for SB to make a single path to the root bridge?

- **A.** 4
- **B.** 8
- **C.** 23
- **D.** 12

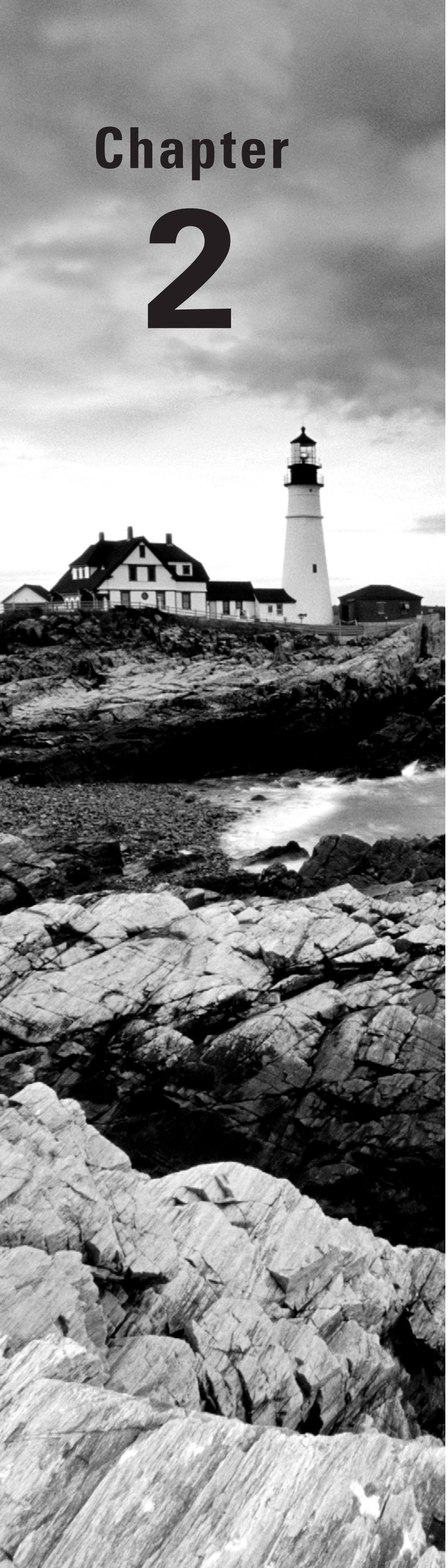
16. Which of the following would put switch interfaces into EtherChannel port number 1, using LACP? (Choose two.)
    - A. `Switch(config)#interface port-channel 1`
    - B. `Switch(config)#channel-group 1 mode active`
    - C. `Switch#interface port-channel 1`
    - D. `Switch(config-if)#channel-group 1 mode active`

17. Which two commands would guarantee your switch to be the root bridge for VLAN 30? (Choose two.)
    - A. `spanning-tree vlan 30 priority 0`
    - B. `spanning-tree vlan 30 priority 16384`
    - C. `spanning-tree vlan 30 root guarantee`
    - D. `spanning-tree vlan 30 root primary`

18. Why does Cisco use its proprietary extension of PVST+ with STP and RSTP?
    - A. Root bridge placement enables faster convergence as well as optimal path determination.
    - B. Non-root bridge placement clearly enables faster convergence as well as optimal path determination.
    - C. PVST+ allows for faster discarding of non-IP frames.
    - D. PVST+ is actually an IEEE standard called 802.1w.

19. Which are states in 802.1d? (Choose all that apply.)
    - A. Blocking
    - B. Discarding
    - C. Listening
    - D. Learning
    - E. Forwarding
    - F. Alternate

20. Which of the following are roles in STP? (Choose all that apply.)
    - A. Blocking
    - B. Discarding
    - C. Root
    - D. Non-designated
    - E. Forwarding
    - F. Designated



Chapter 2

Managing Cisco Devices

THE FOLLOWING ICND2 EXAM TOPICS ARE COVERED IN THIS CHAPTER:

✓ **IP Routing Technologies**

- Describe the boot process of Cisco IOS routers
  - POST
  - Router bootup process
- Manage Cisco IOS Files
  - Boot preferences
  - Cisco IOS image(s)
  - Licensing
    - Show license
    - Change license

Here in Chapter 2, I'm going to show you how to manage Cisco routers on an internetwork. The Internetwork Operating System (IOS) and configuration files reside in different locations in a Cisco device, so it's really important to understand both where these files are located and how they work.

You'll be learning about the main components of a router, the router boot sequence, and the configuration register, including how to use the configuration register for password recovery.

Finally, I'll cover how to verify licenses on the ISRG2 routers as well as how to install a permanent license and configure evaluation features in the latest universal images.

NOTE To find up-to-the-minute updates for this chapter, please see `www.lammle.com/forum` or the book's web page at `www.sybex.com`.

The Internal Components of a Cisco Router

To configure and troubleshoot a Cisco internetwork, you need to know the major components of Cisco routers and understand what each one does. Table 2.1 describes the major Cisco router components.

TABLE 2.1 Cisco router components

Component	Description
Bootstrap	Stored in the microcode of the ROM, the bootstrap is used to bring a router up during initialization. It will boot the router and then load the IOS.
POST (power-on self-test)	Stored in the microcode of the ROM, the POST is used to check the basic functionality of the router hardware and determines which interfaces are present.

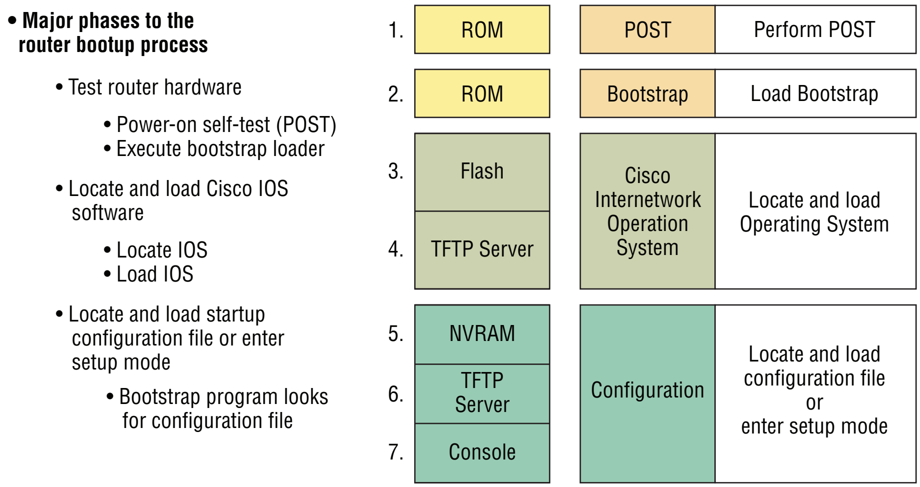
Component	Description
ROM monitor	Stored in the microcode of the ROM, the ROM monitor is used for manufacturing, testing, and troubleshooting. In older routers it could load what was called a mini-IOS.
Mini-IOS	Called the RXBOOT or bootloader by Cisco, the mini-IOS is a small IOS in ROM that can be used to bring up an interface and load a Cisco IOS into flash memory. The mini-IOS can also perform a few other maintenance operations, but not many.
RAM (random access memory)	Used to hold packet buffers, ARP caches, routing tables, and also the software and data structures that allow the router to function. Running-config is stored in RAM, and most routers expand the IOS from flash into RAM upon boot.
ROM (read-only memory)	Used to start and maintain the router. Holds the POST and the bootstrap program as well as the mini-IOS.
Flash memory	Stores the Cisco IOS by default. Flash memory is not erased when the router is reloaded. It is EEPROM (electronically erasable programmable read-only memory) created by Intel.
NVRAM (nonvolatile RAM)	Used to hold the router and switch configuration. NVRAM is not erased when the router or switch is reloaded. Does not store an IOS. The configuration register is stored in NVRAM.
Configuration register	Used to control how the router boots up. This value can be found as the last line of the `show version` command output and by default is set to 0x2102, which tells the router to load the IOS from flash memory as well as to load the configuration from NVRAM.

The Router Boot Sequence

When a router boots up, it performs a series of steps, called the *boot sequence*, to test the hardware and load the necessary software. The boot sequence consists of the following steps, as shown in Figure 2.1:

1. The router performs a POST. The POST tests the hardware to verify that all components of the device are operational and present. For example, the POST checks for the different interfaces on the router. The POST is stored in and run from *ROM (read-only memory)*.

FIGURE 2.1 Router bootup process


• Major phases to the router bootup process
• Test router hardware
• Power-on self-test (POST)
• Execute bootstrap loader
• Locate and load Cisco IOS software
• Locate IOS
• Load IOS
• Locate and load startup configuration file or enter setup mode
• Bootstrap program looks for configuration file
1. ROM
POST
Perform POST
2. ROM
Bootstrap
Load Bootstrap
3. Flash
4. TFTP Server
Cisco Internetwork Operation System
Locate and load Operating System
5. NVRAM
6. TFTP Server
7. Console
Configuration
Locate and load configuration file or enter setup mode

2. The bootstrap then looks for and loads the Cisco IOS software. The bootstrap is a program in ROM that is used to execute programs. The bootstrap program is responsible for finding where each IOS program is located and then loading the file. By default, the IOS software is loaded from flash memory in all Cisco routers, but only after the configuration register is read and NVRAM is checked for the boot sequence.

TIP

The default order of an IOS loading from a router is: flash, TFTP server, then ROM.

3. After the IOS is expanded into RAM, the IOS software looks for a valid configuration file stored in NVRAM. This file is called startup-config and is only there if an administrator copies the running-config file into NVRAM.
4. If a startup-config file is in NVRAM, the router will copy this file and place it in RAM and call the file running-config. The router will use this file to run the router. The router should now be operational. If a startup-config file is not in NVRAM, the router will broadcast out any interface that detects carrier detect (CD) for a TFTP host looking for a configuration, and when that fails (typically it will fail—most people won't even realize the router has attempted this process), it will start the setup mode configuration process.

Managing Configuration Register

All Cisco routers have a 16-bit software register that's written into NVRAM. By default, the *configuration register* is set to load the Cisco IOS from *flash memory* and to look for and load the startup-config file from NVRAM. In the following sections, I am going to discuss the configuration register settings and how to use these settings to provide password recovery on your routers.

Understanding the Configuration Register Bits

The 16 bits (2 bytes) of the configuration register are read from 15 to 0, from left to right. The default configuration setting on Cisco routers is 0x2102. This means that bits 13, 8, and 1 are on, as shown in Table 2.2. Notice that each set of 4 bits (called a nibble) is read in binary with a value of 8, 4, 2, 1.

TABLE 2.2 The configuration register bit numbers

Configuration Register			**2**					**1**				**0**			**2**	
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Binary	0	0	1	0	0	0	0	1	0	0	0	0	0	0	1	0

NOTE

Add the prefix *0x* to the configuration register address. The *0x* means that the digits that follow are in hexadecimal.

Table 2.3 lists the software configuration bit meanings. Notice that bit 6 can be used to ignore the NVRAM contents. This bit is used for password recovery—something I'll go over with you soon in the section "Recovering Passwords," later in this chapter.

NOTE

Remember that in hex, the scheme is 0–9 and A–F (A = 10, B = 11, C = 12, D = 13, E = 14, and F = 15). This means that a 210F setting for the configuration register is actually 210(15), or 1111 in binary.

TABLE 2.3 Software configuration meanings

Bit	**Hex**	**Description**
0–3	0x0000–0x000F	Boot field (see Table 2.4).
6	0x0040	Ignore NVRAM contents.
7	0x0080	OEM bit enabled.

TABLE 2.3 Software configuration meanings *(continued)*

Bit	Hex	Description
8	0x101	Break disabled.
10	0x0400	IP broadcast with all zeros.
5, 11–12	0x0800–0x1000	Console line speed.
13	0x2000	Boot default ROM software if network boot fails.
14	0x4000	IP broadcasts do not have net numbers.
15	0x8000	Enable diagnostic messages and ignore NVRAM contents.

The boot field, which consists of bits 0–3 in the configuration register (the last four bits), controls the router boot sequence and locates the Cisco IOS. Table 2.4 describes the boot field bits.

TABLE 2.4 The boot field (configuration register bits 00–03)

Boot Field	Meaning	Use
00	ROM monitor mode	To boot to ROM monitor mode, set the configuration register to 2100. You must manually boot the router with the `b` command. The router will show the `rommon>` prompt.
01	Boot image from ROM	To boot the mini-IOS image stored in ROM, set the configuration register to 2101. The router will show the `Router(boot)>` prompt. The mini-IOS is not available in all routers and is also referred to as RXBOOT.
02–F	Specifies a default boot filename	Any value from 2102 through 210F tells the router to use the `boot` commands specified in NVRAM.

Checking the Current Configuration Register Value

You can see the current value of the configuration register by using the `show version` command (`sh version` or `show ver` for short), as demonstrated here:

```
Router>sh version
Cisco IOS Software, 2800 Software (C2800NM-ADVSECURITYK9-M),
Version 15.1(4)M6, RELEASE SOFTWARE (fc2)
[output cut]
Configuration register is 0x2102
```

The last information given from this command is the value of the configuration register. In this example, the value is 0x2102—the default setting. The configuration register setting of 0x2102 tells the router to look in NVRAM for the boot sequence.

Notice that the `show version` command also provides the IOS version, and in the preceding example, it shows the IOS version as 15.1(4)M6.

NOTE The `show version` command will display system hardware configuration information, the software version, and the names of the boot images on a router.

To change the configuration register, use the `config-register` command from global configuration mode:

```
Router(config)#config-register 0x2142
Router(config)#do sh ver
[output cut]
Configuration register is 0x2102 (will be 0x2142 at next reload)
```

It's important that you are careful when you set the configuration register!

NOTE If you save your configuration and reload the router and it comes up in setup mode, the configuration register setting is probably incorrect.

Boot System Commands

Did you know that you can configure your router to boot another IOS if the flash is corrupted? Well, you can. You can boot all of your routers from a TFTP server, but it's old school, and people just don't do it anymore; it's just for backup in case of failure.

There are some `boot` commands you can play with that will help you manage the way your router boots the Cisco IOS—but please remember, we're talking about the router's IOS here, *not* the router's configuration!

```
Router>en
Router#config t
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#boot ?
  bootstrap  Bootstrap image file
  config     Configuration file
  host       Router-specific config file
  network    Network-wide config file
  system     System image file
```

The `boot` command truly gives you a wealth of options, but first, I'll show you the typical settings that Cisco recommends. So let's get started—the `boot system` command will allow you to tell the router which system IOS file to boot from flash memory. Remember that the router, by default, boots the first system IOS file found in flash. You can change that with the following commands as shown in the output:

```
Router(config)#boot system ?
  WORD   TFTP filename or URL
  flash  Boot from flash memory
  ftp    Boot from a server via ftp
  mop    Boot from a Decnet MOP server
  rcp    Boot from a server via rcp
  rom    Boot from rom
  tftp   Boot from a tftp server
Router(config)#boot system flash c2800nm-advsecurityk9-mz.151-4.M6.bin
```

Notice I could boot from FLASH, FTP, ROM, TFTP or another useless option. The preceding command I used configures the router to boot the IOS listed in it. This is a helpful command for when you load a new IOS into flash and want to test it, or even when you want to totally change which IOS is loading by default.

The next command is considered a fallback routine, but as I said, you can make it a permanent way to have your routers boot from a TFTP host. Personally, I wouldn't necessarily recommend doing this (single point of failure); I'm just showing you that it's possible:

```
Router(config)#boot system tftp ?
  WORD  System image filename
Router(config)#boot system tftp c2800nm-advsecurityk9-mz.151-4.M6.bin?
  Hostname or A.B.C.D  Address from which to download the file
  <cr>
Router(config)#boot system tftp c2800nm-advsecurityk9-mz.151-4.M6.bin 1.1.1.2
Router(config)#
```

As your last recommended fallback option—the one to go to if the IOS in flash doesn't load and the TFTP host does not produce the IOS—load the mini-IOS from ROM like this:

```
Router(config)#boot system rom
Router(config)#do show run | include boot system
boot system flash c2800nm-advsecurityk9-mz.151-4.M6.bin
boot system tftp c2800nm-advsecurityk9-mz.151-4.M6.bin 1.1.1.2
boot system rom
Router(config)#
```

If the preceding configuration is set, the router will try to boot from the TFTP server if flash fails, and if the TFTP boot fails, the mini-IOS will load after six unsuccessful attempts of trying to locate the TFTP server.

In the next section, I'll show you how to load the router into ROM monitor mode so you can perform password recovery.

Recovering Passwords

If you're locked out of a router because you forgot the password, you can change the configuration register to help you get back on your feet. As I said earlier, bit 6 in the configuration register is used to tell the router whether to use the contents of NVRAM to load a router configuration.

The default configuration register value is 0x2102, meaning that bit 6 is off. With the default setting, the router will look for and load a router configuration stored in NVRAM (startup-config). To recover a password, you need to turn on bit 6. Doing this will tell the router to ignore the NVRAM contents. The configuration register value to turn on bit 6 is 0x2142.

Here are the main steps to password recovery:

1. Boot the router and interrupt the boot sequence by performing a break, which will take the router into ROM monitor mode.
2. Change the configuration register to turn on bit 6 (with the value 0x2142).
3. Reload the router.
4. Say "no" to entering Setup mode, the enter privileged mode.
5. Copy the startup-config file to running-config, and don't forget to verify that your interfaces are re-enabled.
6. Change the password.
7. Reset the configuration register to the default value.
8. Save the router configuration.
9. Reload the router (optional).

I'm going to cover these steps in more detail in the following sections. I'll also show you the commands to restore access to ISR series routers.

You can enter ROM monitor mode by pressing Ctrl+Break or Ctrl+Shift+6 then b, during router bootup. But if the IOS is corrupt or missing, if there's no network connectivity available to find a TFTP host, or if the mini-IOS from ROM doesn't load (meaning the default router fallback failed), the router will enter ROM monitor mode by default.

Interrupting the Router Boot Sequence

Your first step is to boot the router and perform a break. This is usually done by pressing the Ctrl+Break key combination when using HyperTerminal (personally, I use SecureCRT or PuTTY) while the router first reboots.

```
System Bootstrap, Version 15.1(4)M6, RELEASE SOFTWARE (fc2)
Copyright (c) 1999 by cisco Systems, Inc.
TAC:Home:SW:IOS:Specials for info
PC = 0xfff0a530, Vector = 0x500, SP = 0x680127b0
C2800 platform with 32768 Kbytes of main memory
PC = 0xfff0a530, Vector = 0x500, SP = 0x80004374
monitor: command "boot" aborted due to user interrupt
rommon 1 >
```

Notice the line `monitor: command "boot" aborted due to user interrupt`. At this point, you will be at the `rommon 1>` prompt, which is called the ROM monitor mode.

Changing the Configuration Register

As I explained earlier, you can change the configuration register from within the IOS by using the `config-register` command. To turn on bit 6, use the configuration register value 0x2142.

NOTE Remember that if you change the configuration register to 0x2142, the startup-config will be bypassed and the router will load into setup mode.

To change the bit value on a Cisco ISR series router, you just enter the following command at the `rommon 1>` prompt:

```
rommon 1 >confreg 0x2142
You must reset or power cycle for new config to take effect
rommon 2 >reset
```

Reloading the Router and Entering Privileged Mode

At this point, you need to reset the router like this:

- From the ISR series router, type **`I`** (for initialize) or **`reset`**.
- From an older series router, type **`I`**.

The router will reload and ask if you want to use setup mode (because no startup-config is used). Answer no to entering setup mode, press Enter to go into user mode, and then type **`enable`** to go into privileged mode.

Viewing and Changing the Configuration

Now you're past the point where you would need to enter the user-mode and privileged-mode passwords in a router. Copy the startup-config file to the running-config file:

```
copy startup-config running-config
```

Or use the shortcut:

```
copy start run
```

The configuration is now running in *random access memory (RAM)*, and you're in privileged mode, meaning that you can now view and change the configuration. But you can't view the enable-secret setting for the password since it is encrypted. To change the password, do this:

```
config t
enable secret todd
```

Resetting the Configuration Register and Reloading the Router

After you're finished changing passwords, set the configuration register back to the default value with the `config-register` command:

```
config t
config-register 0x2102
```

It's important to remember to enable your interfaces after copying the configuration from NVRAM to RAM.

Finally, save the new configuration with a `copy running-config startup-config` and use `reload` to reload the router.

NOTE If you save your configuration and reload the router and it comes up in setup mode, the configuration register setting is probably incorrect.

To sum this up, we now have Cisco's suggested IOS backup routine configured on our router: flash, TFTP host, ROM.

Visit ccna.gg/ch2/b for a companion MicroNugget from CBT Nuggets.

Backing Up and Restoring the Cisco IOS

Before you upgrade or restore a Cisco IOS, you really should copy the existing file to a *TFTP host* as a backup just in case the new image crashes and burns.

And you can use any TFTP host to accomplish this. By default, the flash memory in a router is used to store the Cisco IOS. In the following sections, I'll describe how to check the amount of flash memory, how to copy the Cisco IOS from flash memory to a TFTP host, and how to copy the IOS from a TFTP host to flash memory.

But before you back up an IOS image to a network server on your intranet, you've got to do these three things:

- Make sure you can access the network server.
- Ensure that the network server has adequate space for the code image.
- Verify the file naming and path requirements.

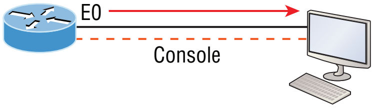
You can connect your laptop or workstation's Ethernet port directly to a router's Ethernet interface, as shown in Figure 2.2.

FIGURE 2.2 Copying an IOS from a router to a TFTP host

Copy the IOS to a TFTP host.

Router# copy flash tftp

- IP address of the TFTP server
- IOS filename



```
RouterX#copy flash tftp:
Source filename [] ?c2800nm-ipbase-mz.124-5a.bin
Address or name of remote host [] ? 10.1.1.1
Destination filename [c2800nm-ipbase-mz.124-5a.bin] [enter]
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!<output omitted>
12094416 bytes copied in 98.858 secs (122341 bytes/sec)
RouterX#
```

- TFTP server software must be running on the PC.
- The PC must be on the same subnet as the router's E0 interface.
- The `copy flash tftp` command must be supplied the IP address of the PC.

You need to verify the following before attempting to copy the image to or from the router:

- TFTP server software must be running on the laptop or workstation.
- The Ethernet connection between the router and the workstation must be made with a crossover cable.
- The workstation must be on the same subnet as the router's Ethernet interface.
- The `copy flash tftp` command must be supplied the IP address of the workstation if you are copying from the router flash.
- And if you're copying "into" flash, you need to verify that there's enough room in flash memory to accommodate the file to be copied.

Verifying Flash Memory

Before you attempt to upgrade the Cisco IOS on your router with a new IOS file, it's a good idea to verify that your flash memory has enough room to hold the new image. You verify the amount of flash memory and the file or files being stored in flash memory by using the `show flash` command (`sh flash` for short):

```
Router#sh flash
-#- --length-- -----date/time------ path
1    45392400 Apr 14 2013 05:31:44 +00:00 c2800nm-advsecurityk9-mz.151-4.M6.bin


18620416 bytes available (45395968 bytes used)
```

There are about 45 MB of flash used, but there still about 18 MB available. If you want to copy a file into flash that is more than 18 MB in size, the router will ask you if you want to erase flash. Be careful here!

NOTE

The `show flash` command will display the amount of memory consumed by the current IOS image, as well as, tell you if there's enough room available to hold both current and new images. You should know that if there's not enough room for both the old and new image you want to load, the old image will be erased!

The amount of RAM and flash is actually easy to tally using the `show version` command on routers:

```
Router#show version
[output cut]
System returned to ROM by power-on
System image file is "flash:c2800nm-advsecurityk9-mz.151-4.M6.bin"
[output cut]
Cisco 2811 (revision 1.0) with 249856K/12288K bytes of memory.
Processor board ID FTX1049A1AB
2 FastEthernet interfaces
2 Serial(sync/async) interfaces
1 Virtual Private Network (VPN) Module
DRAM configuration is 64 bits wide with parity enabled.
239K bytes of non-volatile configuration memory.
62720K bytes of ATA CompactFlash (Read/Write)
```

The first highlighted line shows us that this router has about 256 MB of RAM, and you can see that the amount of flash shows up on the last line. By estimating up, we get the amount of flash to 64 MB.

Notice that the filename in this example is `c2800nm-advsecurityk9-mz.151-4.M6.bin`. The main difference in the output of the `show flash` and `show version` commands is that the `show flash` command displays all files in flash memory and the `show version` command shows the actual name of the file used to run the router and the location from which it was loaded, which is flash memory.

Backing Up the Cisco IOS

To back up the Cisco IOS to a TFTP server, you use the `copy flash tftp` command. It's a straightforward command that requires only the source filename and the IP address of the TFTP server.

The key to success in this backup routine is to make sure you've got good, solid connectivity to the TFTP server. Check this by pinging the TFTP device from the router console prompt like this:

```
Router#ping 1.1.1.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 1.1.1.2, timeout
  is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max
  = 4/4/8 ms
```

After you ping the TFTP server to make sure that IP is working, you can use the `copy flash tftp` command to copy the IOS to the TFTP server as shown next:

```
Router#copy flash tftp
Source filename []?c2800nm-advsecurityk9-mz.151-4.M6.bin
Address or name of remote host []?1.1.1.2
Destination filename [c2800nm-advsecurityk9-mz.151-4.M6.bin]?[enter]
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
45395968 bytes copied in 123.724 secs (357532 bytes/sec)
Router#
```

Just copy the IOS filename from either the `show flash` or `show version` command and then paste it when prompted for the source filename.

In the preceding example, the contents of flash memory were copied successfully to the TFTP server. The address of the remote host is the IP address of the TFTP host, and the source filename is the file in flash memory.

WARNING

Many newer Cisco routers have removable memory. You may see names for this memory such as flash0:, in which case the command in the preceding example would be `copy flash0: tftp:`. Alternately, you may see it as `usbflash0:`.

Restoring or Upgrading the Cisco Router IOS

What happens if you need to restore the Cisco IOS to flash memory to replace an original file that has been damaged or if you want to upgrade the IOS? You can download the file from a TFTP server to flash memory by using the `copy tftp flash` command. This command requires the IP address of the TFTP host and the name of the file you want to download.

But before you begin, make sure the file you want to place in flash memory is in the default TFTP directory on your host. When you issue the command, TFTP won't ask you where the file is, so if the file you want to use isn't in the default directory of the TFTP host, this just won't work.

```
Router#copy tftp flash
Address or name of remote host []?1.1.1.2
Source filename []?c2800nm-advsecurityk9-mz.151-4.M6.bin
Destination filename [c2800nm-advsecurityk9-mz.151-4.M6.bin]?[enter]
%Warning: There is a file already existing with this name
Do you want to over write? [confirm][enter]
Accessing tftp://1.1.1.2/ c2800nm-advsecurityk9-mz.151-4.M6.bin...
Loading c2800nm-advsecurityk9-mz.151-4.M6.bin from 1.1.1.2 (via
   FastEthernet0/0): !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
[OK - 21710744 bytes]


45395968 bytes copied in 82.880 secs (261954 bytes/sec)
Router#
```

In the preceding example, I copied the same file into flash memory, so it asked me if I wanted to overwrite it. Remember that we are "playing" with files in flash memory. If I had just corrupted my file by overwriting it, I won't know for sure until I reboot the router. Be careful with this command! If the file is corrupted, you'll need to do an IOS-restore from ROM monitor mode.

If you are loading a new file and you don't have enough room in flash memory to store both the new and existing copies, the router will ask to erase the contents of flash memory before writing the new file into flash memory, and if you are able to copy the IOS without erasing the old version, then make sure you remember to use the `boot system flash:`*`ios-file`* command.

NOTE As I demonstrated in the Real World Scenario, a Cisco router can become a TFTP server host for a router system image that's run in flash memory. The global configuration command is `tftp-server flash:ios-file`.

Real World Scenario

It's Monday Morning and You Just Upgraded Your IOS

You came in early to work to upgrade the IOS on your router. After the upgrade, you reload the router and the router now shows the `rommon>` prompt.

It seems that you're about to have a bad day! This is what I call an RGE: a resume-generating event! So, now what do you do? Just keep calm and chive on! Follow these steps to save your job:

```
rommon 1 > tftpdnld


Missing or illegal ip address for variable IP_ADDRESS
Illegal IP address.


usage: tftpdnld [-hr]
  Use this command for disaster recovery only to recover an image via TFTP.
  Monitor variables are used to set up parameters for the transfer.
  (Syntax: "VARIABLE_NAME=value" and use "set" to show current variables.)
  "ctrl-c" or "break" stops the transfer before flash erase begins.


  The following variables are REQUIRED to be set for tftpdnld:
            IP_ADDRESS: The IP address for this unit
        IP_SUBNET_MASK: The subnet mask for this unit
       DEFAULT_GATEWAY: The default gateway for this unit
           TFTP_SERVER: The IP address of the server to fetch from
             TFTP_FILE: The filename to fetch


  The following variables are OPTIONAL:
[unneeded output cut]
rommon 2 >set IP_Address:1.1.1.1
rommon 3 >set IP_SUBNET_MASK:255.0.0.0
rommon 4 >set DEFAULT_GATEWAY:1.1.1.2
rommon 5 >set TFTP_SERVER:1.1.1.2
rommon 6 >set TFTP_FILE: flash:c2800nm-advipservicesk9-mz.124-12.bin
rommon 7 >tftpdnld
```

From here you can see the variables you need to configure using the `set` command; be sure you use ALL_CAPS with these commands as well as underscore (_).From here, you need to set the IP address, mask, and default gateway of your router, then the IP address of the TFTP host, which in this example is a directly connected router that I made a TFTP server with this command:

```
Router(config)#tftp-server flash:c2800nm-advipservicesk9-mz.124-12.bin
```

And finally, you set the IOS filename of the file on your TFTP server. Whew! Job saved.

There is one other way you can restore the IOS on a router, but it takes a while. You can use what is called the `Xmodem` protocol to actually upload an IOS file into flash memory through the console port. You'd use the `Xmodem` through the console port procedure if you had no network connectivity to the router or switch.

Using the Cisco IOS File System (Cisco IFS)

Visit ccna .gg/ch2/a for a companion MicroNugget from CBT Nuggets.

Cisco has created a file system called Cisco IFS that allows you to work with files and directories just as you would from a Windows DOS prompt. The commands you use are `dir`, `copy`, `more`, `delete`, `erase` or `format`, `cd` and `pwd`, and `mkdir` and `rmdir`.

Working with IFS gives you the ability to view all files, even those on remote servers. And you definitely want to find out if an image on one of your remote servers is valid before you copy it, right? You also need to know how big it is—size matters here! It's also a really good idea to take a look at the remote server's configuration and make sure it's all good before loading that file on your router.

It's very cool that IFS makes the file system user interface universal—it's not platform specific anymore. You now get to use the same syntax for all your commands on all of your routers, no matter the platform!

Sound too good to be true? Well, it kind of is because you'll find out that support for all commands on each file system and platform just isn't there. But it's really no big deal since various file systems differ in the actions they perform; the commands that aren't relevant to a particular file system are the very ones that aren't supported on that file system. Be assured that any file system or platform will fully support all the commands you need to manage it.

Another cool IFS feature is that it cuts down on all those obligatory prompts for a lot of the commands. If you want to enter a command, all you have to do is type all the necessary info straight into the command line—no more jumping through hoops of prompts! So, if you want to copy a file to an FTP server, all you'd do is first indicate where the desired source file is on your router, pinpoint where the destination file is to be on the FTP server, determine the username and password you're going to use when you want to connect to that server, and type it all in on one line—sleek! And for those of you resistant to change, you can still have the router prompt you for all the information it needs and enjoy entering a more elegantly minimized version of the command than you did before.

But even in spite of all this, your router might still prompt you—even if you did everything right in your command line. It comes down to how you've got the `file prompt` command configured and which command you're trying to use. But no worries—if that happens, the default value will be entered right there in the command, and all you have to do is hit Enter to verify the correct values.

IFS also lets you explore various directories and inventory files in any directory you want. Plus, you can make subdirectories in flash memory or on a card, but you only get to do that if you're working on one of the more recent platforms.

And get this—the new file system interface uses URLs to determine the whereabouts of a file. So just as they pinpoint places on the Web, URLs now indicate where files are on your Cisco router, or even on a remote file server! You just type URLs right into your commands to identify where the file or directory is. It's really that easy—to copy a file from one place to another, you simply enter the `copy source-url destination-url` command—sweet! IFS URLs are a tad different than what you're used to though, and there's an array of formats to use that vary depending on where, exactly, the file is that you're after.

We're going to use Cisco IFS commands pretty much the same way that we used the `copy` command in the IOS section earlier:

- For backing up the IOS
- For upgrading the IOS
- For viewing text files

Okay—with all that down, let's take a look at the common IFS commands available to us for managing the IOS. I'll get into configuration files soon, but for now I'm going to get you started with going over the basics used to manage the new Cisco IOS.

`dir` Same as with Windows, this command lets you view files in a directory. Type **`dir`**, hit Enter, and by default you get the contents of the `flash:/` directory output.

`copy` This is one popular command, often used to upgrade, restore, or back up an IOS. But as I said, when you use it, it's really important to focus on the details—what you're copying, where it's coming from, and where it's going to land.

`more` Same as with Unix, this will take a text file and let you look at it on a card. You can use it to check out your configuration file or your backup configuration file. I'll go over it more when we get into actual configuration.

`show file` This command will give you the skinny on a specified file or file system, but it's kind of obscure because people don't use it a lot.

`delete` Three guesses—yep, it deletes stuff. But with some types of routers, not as well as you'd think. That's because even though it whacks the file, it doesn't always free up the space it was using. To actually get the space back, you have to use something called the `squeeze` command too.

`erase/format` Use these with care—make sure that when you're copying files, you say no to the dialog that asks you if you want to erase the file system! The type of memory you're using determines if you can nix the flash drive or not.

`cd/pwd` Same as with Unix and DOS, `cd` is the command you use to change directories. Use the `pwd` command to print (show) the working directory.

`mkdir/rmdir` Use these commands on certain routers and switches to create and delete directories—the `mkdir` command for creation and the `rmdir` command for deletion. Use the `cd` and `pwd` commands to change into these directories.

NOTE The Cisco IFS uses the alternate term `system:running-config` as well as `nvram:startup-config` when copying the configurations on a router, although it is not mandatory that you use this naming convention.

Using the Cisco IFS to Upgrade an IOS

Let's take a look at some of these Cisco IFS commands on my ISR router (1841 series) with a hostname of R1.

We'll start with the `pwd` command to verify our default directory and then use the `dir` command to verify its contents (`flash:/`):

```
R1#pwd
flash:
R1#dir
Directory of flash:/
    1  -rw-    13937472  Dec 20 2006 19:58:18 +00:00  c1841-ipbase-
   mz.124-1c.bin
    2  -rw-        1821  Dec 20 2006 20:11:24 +00:00  sdmconfig-18xx.cfg
    3  -rw-     4734464  Dec 20 2006 20:12:00 +00:00  sdm.tar
    4  -rw-      833024  Dec 20 2006 20:12:24 +00:00  es.tar
    5  -rw-     1052160  Dec 20 2006 20:12:50 +00:00  common.tar
    6  -rw-        1038  Dec 20 2006 20:13:10 +00:00  home.shtml
    7  -rw-      102400  Dec 20 2006 20:13:30 +00:00  home.tar
    8  -rw-      491213  Dec 20 2006 20:13:56 +00:00  128MB.sdf
    9  -rw-     1684577  Dec 20 2006 20:14:34 +00:00  securedesktop-
   ios-3.1.1.27-k9.pkg
   10  -rw-      398305  Dec 20 2006 20:15:04 +00:00  sslclient-win-1.1.0.154.pkg

32071680 bytes total (8818688 bytes free)
```

What we can see here is that we have the basic IP IOS (`c1841-ipbase-mz.124-1c.bin`). Looks like we need to upgrade our 1841. You've just got to love how Cisco puts the IOS type in the filename now! First, let's check the size of the file that's in flash with the `show file` command (`show flash` would also work):

```
R1#show file info flash:c1841-ipbase-mz.124-1c.bin
flash:c1841-ipbase-mz.124-1c.bin:
  type is image (elf) []
  file size is 13937472 bytes, run size is 14103140 bytes
  Runnable image, entry point 0x8000F000, run from ram
```

With a file that size, the existing IOS will have to be erased before we can add our new IOS file (`c1841-advipservicesk9-mz.124-12.bin`), which is over 21 MB. We'll use the `delete` command, but remember, we can play with any file in flash memory and nothing serious will happen until we reboot—that is, if we made a mistake. So obviously, and as I pointed out earlier, we need to be very careful here!

```
R1#delete flash:c1841-ipbase-mz.124-1c.bin
Delete filename [c1841-ipbase-mz.124-1c.bin]?[enter]
Delete flash:c1841-ipbase-mz.124-1c.bin? [confirm][enter]
R1#sh flash
-#- --length-- -----date/time------ path
1         1821 Dec 20 2006 20:11:24 +00:00 sdmconfig-18xx.cfg
2      4734464 Dec 20 2006 20:12:00 +00:00 sdm.tar
3       833024 Dec 20 2006 20:12:24 +00:00 es.tar
4      1052160 Dec 20 2006 20:12:50 +00:00 common.tar
5         1038 Dec 20 2006 20:13:10 +00:00 home.shtml
6       102400 Dec 20 2006 20:13:30 +00:00 home.tar
7       491213 Dec 20 2006 20:13:56 +00:00 128MB.sdf
8      1684577 Dec 20 2006 20:14:34 +00:00 securedesktop-ios-3.1.1.27-k9.pkg
9       398305 Dec 20 2006 20:15:04 +00:00 sslclient-win-1.1.0.154.pkg
22757376 bytes available (9314304 bytes used)
R1#sh file info flash:c1841-ipbase-mz.124-1c.bin
%Error opening flash:c1841-ipbase-mz.124-1c.bin (File not found)
R1#
```

So with the preceding commands, we deleted the existing file and then verified the deletion by using both the `show flash` and `show file` commands. We'll add the new file with the copy command, but again, we need to make sure to be careful because this way isn't any safer than the first method I showed you earlier:

```
R1#copy tftp://1.1.1.2/c1841-advipservicesk9-mz.124-12.bin/ flash:/
    c1841-advipservicesk9-mz.124-12.bin
```

```
Source filename [/c1841-advipservicesk9-mz.124-12.bin/]?[enter]
Destination filename [c1841-advipservicesk9-mz.124-12.bin]?[enter]
Loading /c1841-advipservicesk9-mz.124-12.bin/ from 1.1.1.2 (via
    FastEthernet0/0): !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
[output cut]
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
[OK - 22103052 bytes]
22103052 bytes copied in 72.008 secs (306953 bytes/sec)
R1#sh flash
-#- --length-- -----date/time------ path
1         1821 Dec 20 2006 20:11:24 +00:00 sdmconfig-18xx.cfg
2      4734464 Dec 20 2006 20:12:00 +00:00 sdm.tar
3       833024 Dec 20 2006 20:12:24 +00:00 es.tar
4      1052160 Dec 20 2006 20:12:50 +00:00 common.tar
5         1038 Dec 20 2006 20:13:10 +00:00 home.shtml
6       102400 Dec 20 2006 20:13:30 +00:00 home.tar
7       491213 Dec 20 2006 20:13:56 +00:00 128MB.sdf
8      1684577 Dec 20 2006 20:14:34 +00:00 securedesktop-ios-3.1.1.27-k9.pkg
9       398305 Dec 20 2006 20:15:04 +00:00 sslclient-win-1.1.0.154.pkg
10    22103052 Mar 10 2007 19:40:50 +00:00 c1841-advipservicesk9-mz.124-12.bin
651264 bytes available (31420416 bytes used)
R1#
```

We can also check the file information with the `show file` command:

```
R1#sh file information flash:c1841-advipservicesk9-mz.124-12.bin
flash:c1841-advipservicesk9-mz.124-12.bin:
  type is image (elf) []
  file size is 22103052 bytes, run size is 22268736 bytes
  Runnable image, entry point 0x8000F000, run from ram
```

Remember that the IOS is expanded into RAM when the router boots, so the new IOS will not run until you reload the router.

I really recommend experimenting with the Cisco IFS commands on a router just to get a good feel for them because, as I've said, they can definitely give you some grief if not executed properly!

I mention "safer methods" a lot in this chapter. Clearly, I've caused myself some serious pain by not being careful enough when working in flash memory! I cannot stress this enough—pay attention when messing around with flash memory!

One of the brilliant features of the ISR routers is that they use the physical flash cards that are accessible from the front or back of any router. These typically have a name like `usbflash0:`, so to view the contents you'd type **`dir usbflash0:`**, for example. You can pull these flash cards out, put them in an appropriate slot in your PC and the card will show up as a drive. You can then add, change, and delete files. Just put the flash card back in your router and power up—instant upgrade. Nice!

Licensing

IOS licensing is now done quite differently than it was with previous versions of the IOS. Actually, there was no licensing before the new 15.0 IOS code, just your word and honor, and we can only guess based on how all products are downloaded on the Internet daily how well that has worked out for Cisco!

Starting with the IOS 15.0 code, things are much different—almost too different. I can imagine that Cisco will come back toward the middle on its licensing issues, so that the administration and management won't be as detailed as the new 15.0 code license is now; but you can be the judge of that after reading this section.

A new ISR router is pre-installed with the software images and licenses that you ordered, so as long as you ordered and paid for everything you need, you're set! If not, you can just install another license, which can be a tad tedious at first—enough so that installing a license was made an objective on the Cisco exam! Of course, it can be done, but it definitely requires some effort. As is typical with Cisco, if you spend enough money on their products, they tend to make it easier on you and your administration, and the licensing for the newest IOS is no exception, as you'll soon see.

On a positive note, Cisco provides evaluation licenses for most software packages and features that are supported on the hardware you purchased, and it's always nice to be able to try it out before you buy. Once the temporary license expires after 60 days, you need to acquire a permanent license in order to continue to use the extended features that aren't available in your current version. This method of licensing allows you to enable a router to use different parts of the IOS. So, what happens after 60 days? Well, nothing—back to the honor system for now. This is now called *Right-To-Use (RTU) licensing*, and it probably won't always be available via your honor, but for now it is.

But that's not the best part of the new licensing features. Prior to the 15.0 code release, there were eight different software feature sets for each hardware router type. With the IOS 15.0 code, the packaging is now called a *universal image*, meaning all feature sets are available in one file with all features packed neatly inside. So instead of the pre-15.0 IOS file packages of one image per feature set, Cisco now just builds one universal image that includes all of them in the file. Even so, we still need a different universal image per router model or series, just not a different image for each feature set as we did with previous IOS versions.

To use the features in the IOS software, you must unlock them using the software activation process. Since all features available are inside the universal image already, you can just unlock the features you need as you need them, and of course pay for these features when you determine that they meet your business requirements. All routers come with something called the IP Base licensing, which is the prerequisite for installing all other features.

There are three different technology packages available for purchase that can be installed as additional feature packs on top of the prerequisite IP Base (default), which provides entry-level IOS functionality. These are as follows:

Data: MPLS, ATM, and multiprotocol support

Unified Communications: VoIP and IP telephony

Security: Cisco IOS Firewall, IPS, IPsec, 3DES, and VPN

For example, if you need MPLS and IPsec, you'll need the default IP Base, Data, and Security premium packages unlocked on your router.

To obtain the license, you'll need the unique device identifier (UDI), which has two components: the product ID (PID) and the serial number of the router. The `show license UDI` command provides this information in an output as shown:

```
Router#sh license udi
Device#   PID                    SN              UDI
-----------------------------------------------------------------------------
*0       CISCO2901/K9          FTX1641Y07J     CISCO2901/K9:FTX1641Y07J
```

After the time has expired for your 60-day evaluation period, you can either obtain the license file from the Cisco License Manager (CLM), which is an automated process, or use the manual process through the Cisco Product License Registration portal. Typically only larger companies will use the CLM because you'd need to install software on a server, which then keeps track of all your licenses for you. If you have just a few licenses that you use, you can opt for the manual web browser process found on the Cisco Product License Registration portal and then just add in a few CLI commands. After that, you just basically keep track of putting all the different license features together for each device you manage. Although this sounds like a lot of work, you don't need to perform these steps often. But clearly, going with the CLM makes a lot of sense if you have bunches of licenses to manage because it will put together all the little pieces of licensing for each router in one easy process.

When you purchase the software package with the features that you want to install, you need to permanently activate the software package using your UDI and the *product authorization key (PAK)* that you received with your purchase. This is essentially your receipt acknowledging that you purchased the license. You then need to connect the license with a particular router by combining the PAK and the UDI, which you do online at the Cisco Product License Registration portal (`www.cisco.com/go/license`). If you haven't already registered the license on a different router, and it is valid, Cisco will then email you your permanent license, or you can download it from your account.

But wait! You're still not done. You now need to activate the license on the router. Whew… maybe it's worthwhile to install the CLM on a server after all! Staying with the manual method, you need to make the new license file available to the router either via a USB port on the router or through a TFTP server. Once it's available to the router, you'll use the `license install` command from privileged mode.

Assuming that you copied the file into flash memory, the command would look like something like this:

```
Router#license install ?
  archive:  Install from archive: file system
  flash:    Install from flash: file system
  ftp:      Install from ftp: file system
  http:     Install from http: file system
  https:    Install from https: file system
  null:     Install from null: file system
  nvram:    Install from nvram: file system
  rcp:      Install from rcp: file system
  scp:      Install from scp: file system
  syslog:   Install from syslog: file system
  system:   Install from system: file system
  tftp:     Install from tftp: file system
  tmpsys:   Install from tmpsys: file system
  xmodem:   Install from xmodem: file system
  ymodem:   Install from ymodem: file system
Router#license install flash:FTX1628838P_201302111432454180.lic
Installing licenses from "flash::FTX1628838P_201302111432454180.lic"
Installing...Feature:datak9...Successful:Supported
1/1 licenses were successfully installed
0/1 licenses were existing licenses
0/1 licenses were failed to install
April 12 2:31:19.786: %LICENSE-6-INSTALL: Feature datak9 1.0 was
installed in this device. UDI=CISCO2901/K9:FTX1628838P; StoreIndex=1:Primary
License Storage


April 12 2:31:20.078: %IOS_LICENSE_IMAGE_APPLICATION-6-LICENSE_LEVEL: Module name
=c2800 Next reboot level = datak9 and License = datak9
```

You need to reboot to have the new license take effect. Now that you have your license installed and running, how do you use Right-To-Use licensing to check out new features on your router? Let's look into that now.

Right-To-Use Licenses (Evaluation Licenses)

Originally called evaluation licenses, Right-To-Use (RTU) licenses are what you need when you want to update your IOS to load a new feature but either don't want to wait to get the license or just want to test if this feature will truly meet your business requirements. This makes sense because if Cisco made it complicated to load and check out a feature, they could potentially miss out on a sale! Of course if the feature does work for you, they'll want you to buy a permanent license, but again, this is on the honor system at the time of this writing.

Cisco's license model allows you to install the feature you want without a PAK. The Right-To-Use license works for 60 days before you would need to install your permanent license. To enable the Right-To-Use license you would use the `license boot module` command. The following demonstrates starting the Right-To-Use license on my 2900 series router, enabling the security module named `securityk9`:

```
Router(config)#license boot module c2900 technology-package securityk9
PLEASE READ THE FOLLOWING TERMS CAREFULLY. INSTALLING THE LICENSE OR LICENSE KEY
PROVIDED FOR ANY CISCO PRODUCT FEATURE OR USING
SUCHPRODUCT FEATURE CONSTITUTES YOUR FULL ACCEPTANCE OF THE
FOLLOWING TERMS. YOU MUST NOT PROCEED FURTHER IF YOU ARE NOT WILLING
TO BE BOUND BY ALL THE TERMS SET FORTH HEREIN.
[output cut]
Activation of the software command line interface will be evidence of
your acceptance of this agreement.


ACCEPT? [yes/no]: yes


% use 'write' command to make license boot config take effect on next boot
Feb 12 01:35:45.060: %IOS_LICENSE_IMAGE_APPLICATION-6-LICENSE_LEVEL:
Module name =c2900 Next reboot level = securityk9 and License = securityk9


Feb 12 01:35:45.524: %LICENSE-6-EULA_ACCEPTED: EULA for feature
securityk9 1.0 has been accepted. UDI=CISCO2901/K9:FTX1628838P;
StoreIndex=0:Built-In License Storage
```

Once the router is reloaded, you can use the security feature set. And it is really nice that you don't need to reload the router again if you choose to install a permanent license for this feature. The `show license` command shows the licenses installed on the router:

```
Router#show license
Index 1 Feature: ipbasek9
     Period left: Life time
     License Type: Permanent
     License State: Active, In Use
     License Count: Non-Counted
     License Priority: Medium
Index 2 Feature: securityk9
     Period left: 8 weeks  2 days
     Period Used: 0  minute  0  second
     License Type: EvalRightToUse
     License State: Active, In Use
     License Count: Non-Counted
     License Priority: None
Index 3 Feature: uck9
     Period left: Life time
     License Type: Permanent
     License State: Active, In Use
     License Count: Non-Counted
     License Priority: Medium
Index 4 Feature: datak9
     Period left: Not Activated
     Period Used: 0  minute  0  second
     License Type: EvalRightToUse
     License State: Not in Use, EULA not accepted
     License Count: Non-Counted
     License Priority: None
Index 5 Feature: gatekeeper
 [output cut]
```

You can see in the preceding output that the `ipbasek9` is permanent and the `securityk9` has a license type of `EvalRightToUse`. The `show license feature` command provides the same information as `show license`, but it's summarized into one line as shown in the next output:

```
Router#sh license feature
Feature name    Enforcement  Evaluation  Subscription   Enabled  RightToUse
ipbasek9            no           no          no              yes      no
securityk9          yes          yes         no              no       yes
uck9                yes          yes         no              yes      yes
datak9              yes          yes         no              no       yes
gatekeeper          yes          yes         no              no       yes
SSL_VPN             yes          yes         no              no       yes
ios-ips-update      yes          yes         yes             no       yes
SNASw               yes          yes         no              no       yes
hseck9              yes          no          no              no       no
cme-srst            yes          yes         no              yes      yes
WAAS_Express        yes          yes         no              no       yes
UCVideo             yes          yes         no              no       yes
```

The `show version` command also shows the license information at the end of the command output:

```
Router#show version
[output cut]
License Info:


License UDI:


-------------------------------------------------
Device#   PID                   SN
-------------------------------------------------
*0        CISCO2901/K9          FTX1641Y07J


Technology Package License Information for Module:'c2900'


-----------------------------------------------------------------
Technology    Technology-package           Technology-package
              Current       Type           Next reboot
------------------------------------------------------------------
ipbase        ipbasek9      Permanent      ipbasek9
security      None          None           None
uc            uck9          Permanent      uck9
data          None          None           None


Configuration register is 0x2102
```

The `show version` command shows if the license was activated. Don't forget, you'll need to reload the router to have the license features take effect if the license evaluation is not already active.

Backing Up and Uninstalling the License

It would be a shame to lose your license if it has been stored in flash and your flash files become corrupted. So always back up your IOS license!

If your license has been saved in a location other than flash, you can easily back it up to flash memory via the `license save` command:

```
Router#license save flash:Todd_License.lic
```

The above command will save your current license to flash. You can restore your license with the `license install` command I demonstrated earlier.

There are two steps to uninstalling the license on a router. First, to uninstall the license you need to disable the technology package, using the `no license boot module` command with the keyword `disable` at the end of the command line:

```
Router#license boot module c2900 technology-package securityk9 disable
```

The second step is to clear the license. To achieve this from the router, use the `license clear` command and then remove the license with the `no license boot module` command.

```
Router#license clear securityk9
Router#config t
Router(config)#no license boot module c2900 technology-package securityk9 disable
Router(config)#exit
Router#reload
```

After you run through the preceding commands, the license will be removed from your router.

Here's a summary of the license commands I used in this chapter. These are important commands to have down and you really need to understand these to meet the Cisco objectives:

- `show license` determines the licenses that are active on your system. It also displays a group of lines for each feature in the currently running IOS image along with several status variables related to software activation and licensing, both licensed and unlicensed features.
- `show license feature` allows you to view the technology package licenses and feature licenses that are supported on your router along with several status variables related to software activation and licensing. This includes both licensed and unlicensed features.
- `show license udi` displays the unique device identifier (UDI) of the router, which comprises the product ID (PID) and serial number of the router.
- `show version` displays various information about the current IOS version, including the licensing details at the end of the command's output.
- `license install` *`url`* installs a license key file into a router.
- `license boot module` installs a Right-To-Use license feature on a router.

Summary

You now know how Cisco routers are configured and how to manage those configurations.

This chapter covered the internal components of a router, which included ROM, RAM, NVRAM, and flash.

In addition, I covered what happens when a router boots and which files are loaded at that time. The configuration register tells the router how to boot and where to find files. You learned how to change and verify the configuration register settings for password recovery purposes. I also showed you how to manage these files using the CLI and IFS.

Finally, the chapter covered licensing with the new 15.0 code, including how to install a permanent license and Right-To-Use license to install features for 60 days. I also showed you the verification commands used to see what licenses are installed and to verify their status.

Exam Essentials

Define the Cisco router components. Describe the functions of the bootstrap, POST, ROM monitor, mini-IOS, RAM, ROM, flash memory, NVRAM, and the configuration register.

Identify the steps in the router boot sequence. The steps in the boot sequence are: POST, loading the IOS, and copying the startup configuration from NVRAM to RAM.

Understand configuration register commands and settings. The 0x2102 setting is the default on all Cisco routers and tells the router to look in NVRAM for the boot sequence. 0x2101 tells the router to boot from ROM, and 0x2142 tells the router not to load the startup-config in NVRAM to provide password recovery.

Perform password recovery. The steps in the password recovery process are: interrupt the router boot sequence, change the configuration register, reload the router and enter privileged mode, copy the startup-config file to running-config and verify that your interfaces are re-enabled, change/set the password, save the new configuration, reset the configuration register, and reload the router.

Back up an IOS image. By using the privileged-mode command `copy flash tftp`, you can back up a file from flash memory to a TFTP (network) server.

Restore or upgrade an IOS image. By using the privileged-mode command `copy tftp flash`, you can restore or upgrade a file from a TFTP (network) server to flash memory.

Describe best practices to prepare to back up an IOS image to a network server. Make sure that you can access the network server, ensure that the network server has adequate space for the code image, and verify the file naming and path requirement.

Understand and use Cisco IFS file system management commands. The commands to use are `dir`, `copy`, `more`, `delete`, `erase` or `format`, `cd` and `pwd`, and `mkdir` and `rmdir`, as well as `system:running-config` and `nvram:startup-config`.

Remember how to install a permanent and Right-To-Use license. To install a permanent license on a router, use the `install license` *`url`* command. To install an evaluation feature, use the `license boot module` command.

Remember the verification commands used for licensing in the new ISR G2 routers. The `show license` command determines the licenses that are active on your system. The `show license feature` command allows you to view the technology package licenses and feature licenses that are supported on your router. The `show license udi` command displays the unique device identifier (UDI) of the router, which comprises the product ID (PID) and serial number of the router, and the `show version` command displays various information about the current IOS version, including the licensing details at the end of the command's output.

Written Labs

The answers to these labs can be found in Appendix A, "Answers to Written Labs."

In this section, you'll complete the following labs to make sure you've got the information and concepts contained within them fully dialed in:

Lab 2.1: IOS Management

Lab 2.2: Router Memory

Written Lab 2.1: IOS Management

Write the answers to the following questions:

1. What is the command to copy a Cisco IOS to a TFTP server?
2. What do you set the configuration register setting to in order to boot the mini-IOS in ROM?
3. What is the configuration register setting to tell the router to look in NVRAM for the boot sequence?
4. What do you set the configuration register setting to in order to boot to ROM monitor mode?
5. What is used with a PAK to generate a license file?
6. What is the configuration register setting for password recovery?
7. Which command can change the location from which the system loads the IOS?
8. What is the first step of the router boot sequence?
9. What command can you use to upgrade a Cisco IOS?
10. Which command determines the licenses that are active on your system?

Written Lab 2.2: Router Memory

Identify the location in a router where each of the following files is stored by default.

1. Cisco IOS
2. Bootstrap
3. Startup configuration
4. POST routine
5. Running configuration
6. ARP cache
7. Mini-IOS
8. ROM monitor
9. Routing tables
10. Packet buffers

Hands-on Labs

To complete the labs in this section, you need at least one router (three would be best) and at least one PC running as a TFTP server. TFTP server software must be installed and running on the PC. For these labs, it is also assumed that your PC and the router(s) are connected together with a switch or hub and that all interfaces (PC NIC and router interfaces) are in the same subnet. You can alternately connect the PC directly to the router or connect the routers directly to one another (use a crossover cable in that case). Remember that the labs listed here were created for use with real routers but can easily be used with the LammleSim IOS version or Cisco's Packet Tracer program.

Here is a list of the labs in this chapter:

Lab 2.1: Backing Up Your Router IOS

Lab 2.2: Upgrading or Restoring Your Router IOS

Hands-on Lab 2.1: Backing Up Your Router IOS

In this lab, we'll be backing up the IOS from flash to a TFTP host.

1. Log into your router and go into privileged mode by typing **`en`** or **`enable`**.
2. Make sure you can connect to the TFTP server that is on your network by pinging the IP address from the router console.
3. Type **`show flash`** to see the contents of flash memory.
4. Type **`show version`** at the router privileged-mode prompt to get the name of the IOS currently running on the router. If there is only one file in flash memory, the `show flash` and `show version` commands show the same file. Remember that the `show version` command shows you the file that is currently running and the `show flash` command shows you all of the files in flash memory.
5. Once you know you have good Ethernet connectivity to the TFTP server and you also know the IOS filename, back up your IOS by typing **`copy flash tftp`**. This command tells the router to copy a specified file from flash memory (this is where the IOS is stored by default) to a TFTP server.
6. Enter the IP address of the TFTP server and the source IOS filename. The file is now copied and stored in the TFTP server's default directory.

Hands-on Lab 2.2: Upgrading or Restoring Your Router IOS

In this lab, we'll be copying an IOS from a TFTP host to flash memory.

1. Log into your router and go into privileged mode by typing **`en`** or **`enable`**.
2. Make sure you can connect to the TFTP server by pinging the IP address of the server from the router console.

3. Once you know you have good Ethernet connectivity to the TFTP server, type the **`copy tftp flash`** command.
4. Confirm that the router will not function during the restore or upgrade by following the prompts provided on the router console. It is possible this prompt may not occur.
5. Enter the IP address of the TFTP server.
6. Enter the name of the IOS file you want to restore or upgrade.
7. Confirm that you understand the contents of flash memory will be erased if there is not enough room in flash to store the new image.
8. Watch in amazement as your IOS is deleted out of flash memory and your new IOS is copied to flash memory.

If the file that was in flash memory is deleted but the new version wasn't copied to flash memory, the router will boot from ROM monitor mode. You'll need to figure out why the copy operation did not take place.

Review Questions

NOTE The following questions are designed to test your understanding of this chapter's material. For more information on how to get additional questions, please see this book's introduction.

The answers to these questions can be found in Appendix B, "Answers to Review Questions."

1. What does the command `confreg 0x2142` provide?
   - **A.** It is used to restart the router.
   - **B.** It is used to bypass the configuration in NVRAM.
   - **C.** It is used to enter ROM monitor mode.
   - **D.** It is used to view the lost password.
2. Which command will copy the IOS to a backup host on your network?
   - **A.** `transfer IOS to 172.16.10.1`
   - **B.** `copy run start`
   - **C.** `copy tftp flash`
   - **D.** `copy start tftp`
   - **E.** `copy flash tftp`
3. What command is used to permanently install a license on an ISR2 router?
   - **A.** `install license`
   - **B.** `license install`
   - **C.** `boot system license`
   - **D.** `boot license module`
4. You type the following into the router and reload. What will the router do?

```
Router(config)#boot system flash c2800nm-advsecurityk9-mz.151-4.M6.bin
Router(config)#config-register 0x2101
Router(config)#do sh ver
[output cut]
Configuration register is 0x2102 (will be 0x2101 at next reload)
```

   - **A.** The router will expand and run the `c2800nm-advsecurityk9-mz.151-4.M6.bin` IOS from flash memory.
   - **B.** The router will go into setup mode.
   - **C.** The router will load the mini-IOS from ROM.
   - **D.** The router will enter ROM monitor mode.

5. A network administrator wants to upgrade the IOS of a router without removing the image currently installed. What command will display the amount of memory consumed by the current IOS image and indicate whether there is enough room available to hold both the current and new images?
   - **A.** `show version`
   - **B.** `show flash`
   - **C.** `show memory`
   - **D.** `show buffers`
   - **E.** `show running-config`

6. The corporate office sends you a new router to connect, but upon connecting the console cable, you see that there is already a configuration on the router. What should be done before a new configuration is entered in the router?
   - **A.** RAM should be erased and the router restarted.
   - **B.** Flash should be erased and the router restarted.
   - **C.** NVRAM should be erased and the router restarted.
   - **D.** The new configuration should be entered and saved.

7. Which command loads a new version of the Cisco IOS into a router?
   - **A.** `copy flash ftp`
   - **B.** `copy nvram flash`
   - **C.** `copy flash tftp`
   - **D.** `copy tftp flash`

8. Which command will show you the IOS version running on your router?
   - **A.** `sh IOS`
   - **B.** `sh flash`
   - **C.** `sh version`
   - **D.** `sh protocols`

9. What should the configuration register value be after you successfully complete the password recovery procedure and return the router to normal operation?
   - **A.** 0x2100
   - **B.** 0x2101
   - **C.** 0x2102
   - **D.** 0x2142

10. You save the configuration on a router with the `copy running-config startup-config` command and reboot the router. The router, however, comes up with a blank configuration. What can the problem be?
    - **A.** You didn't boot the router with the correct command.
    - **B.** NVRAM is corrupted.
    - **C.** The configuration register setting is incorrect.
    - **D.** The newly upgraded IOS is not compatible with the hardware of the router.
    - **E.** The configuration you saved is not compatible with the hardware.

11. Which command will install a Right-To-Use license so you can use an evaluation version of a feature?
    - **A.** `install Right-To-Use license feature` *`feature`*
    - **B.** `install temporary feature` *`feature`*
    - **C.** `license install feature`
    - **D.** `license boot module`

12. Which command determines the licenses that are active on your system along with several status variables?
    - **A.** `show license`
    - **B.** `show license feature`
    - **C.** `show license udi`
    - **D.** `show version`

13. Which command allows you to view the technology package licenses and feature licenses that are supported on your router along with several status variables?
    - **A.** `show license`
    - **B.** `show license feature`
    - **C.** `show license udi`
    - **D.** `show version`

14. Which command displays the unique device identifier that comprises the product ID and serial number of the router?
    - **A.** `show license`
    - **B.** `show license feature`
    - **C.** `show license udi`
    - **D.** `show version`

15. Which command displays various information about the current IOS version, including the licensing details at the end of the command's output?

A. `show license`

B. `show license feature`

C. `show license udi`

D. `show version`

16. Which command backs up your license to flash memory?

A. `copy tftp flash`

B. `save license flash`

C. `license save flash`

D. `copy license flash`

17. Which command displays the configuration register setting?

A. `show ip route`

B. `show boot version`

C. `show version`

D. `show flash`

18. What two steps are needed to remove a license from a router? (Choose two.)

A. Use the `erase flash:license` command.

B. Reload the system.

C. Use the `license boot` command with the `disable` variable at the end of the command line.

D. Clear the license with the `license clear` command.

19. You have your laptop directly connected into a router's Ethernet port. Which of the following are among the requirements for the `copy flash tftp` command to be successful? (Choose three.)

A. TFTP server software must be running on the router.

B. TFTP server software must be running on your laptop.

C. The Ethernet cable connecting the laptop directly into the router's Ethernet port must be a straight-through cable.

D. The laptop must be on the same subnet as the router's Ethernet interface.

E. The `copy flash tftp` command must be supplied the IP address of the laptop.

F. There must be enough room in the flash memory of the router to accommodate the file to be copied.

20. The configuration register setting of 0x2102 provides what function to a router?

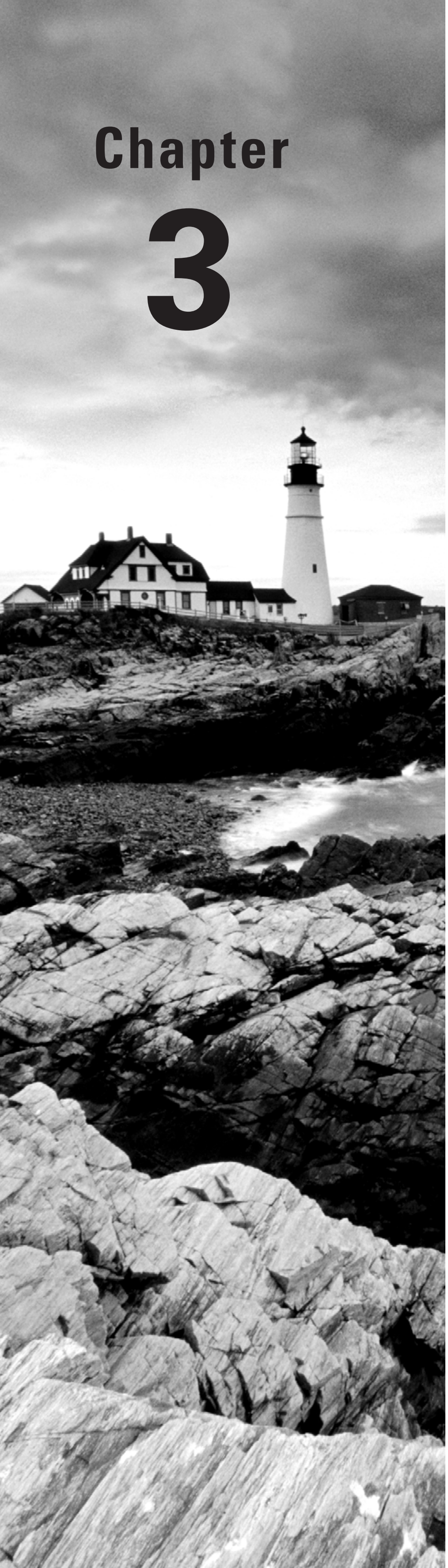
A. Tells the router to boot into ROM monitor mode

B. Provides password recovery

C. Tells the router to look in NVRAM for the boot sequence

D. Boots the IOS from a TFTP server

E. Boots an IOS image stored in ROM



Chapter

3

IP Services

THE FOLLOWING ICND2 EXAM TOPICS ARE COVERED IN THIS CHAPTER:

✓ **IP Services**

- Recognize high availability (FHRP)
  - VRRP
  - HSRP
  - GLBP
- Configure and verify Syslog
  - Utilize Syslog Output
- Describe SNMP v2 & v3

✓ **Troubleshooting**

- Utilize netflow data
- Monitor NetFlow statistics

In this chapter, I'm going to show you how to integrate redundancy and load-balancing features into your network elegantly with the routers that you likely have already. Acquiring some overpriced load-balancing device just isn't always necessary because knowing how to properly configure and use Hot Standby Router Protocol (HSRP), as well as the nonproprietary Virtual Router Redundancy Protocol (VRRP), can often meet your needs instead. You'll soon see that while these technologies are very similar in function, they operate slightly differently. I'll also show you how to configure and use Gateway Load Balancing Protocol (GLBP), which allows you to use up to four routers for an effective load-balancing solution within your first hop redundancy protocols (FHRPs).

I know we discussed syslog in earlier CCENT chapters, but I'm going to delve into it more in-depth in this chapter, as well as look at Simple Network Management Protocol (SNMP) and the type of alerts sent to the network management station (NMS). In addition, we'll cover Cisco NetFlow and how it works in an internetwork.

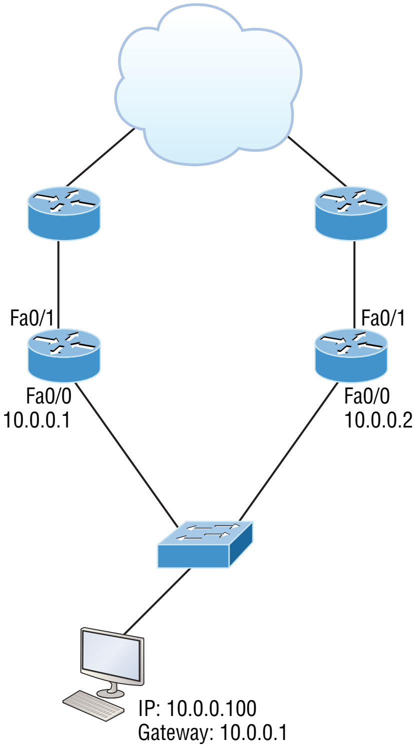
NOTE To find up-to-the-minute updates for this chapter, please see `www.lammle.com/forum` or the book's web page at `www.sybex.com`.

Client Redundancy Issues

If you're wondering how you can possibly configure a client to send data off its local link when its default gateway router has gone down, you've targeted a key issue because the answer is that usually, you can't! Most host operating systems just don't allow you to change data routing. Sure, if a host's default gateway router goes down, the rest of the network will still converge, but it won't share that information with the hosts. Take a look at Figure 3.1 to see what I am talking about. There are actually two routers available to forward data for the local subnet, but the hosts know about only one of them. They learn about this router when you provide them with the default gateway either statically or through DHCP.

This begs the question: Is there another way to use the second active router? The answer is a bit complicated, but bear with me. There is a feature that's enabled by default on Cisco routers called Proxy Address Resolution Protocol (Proxy ARP). Proxy ARP enables hosts, which have no knowledge of routing options, to obtain the MAC address of a gateway router that can forward packets for them.

FIGURE 3.1 Default gateway

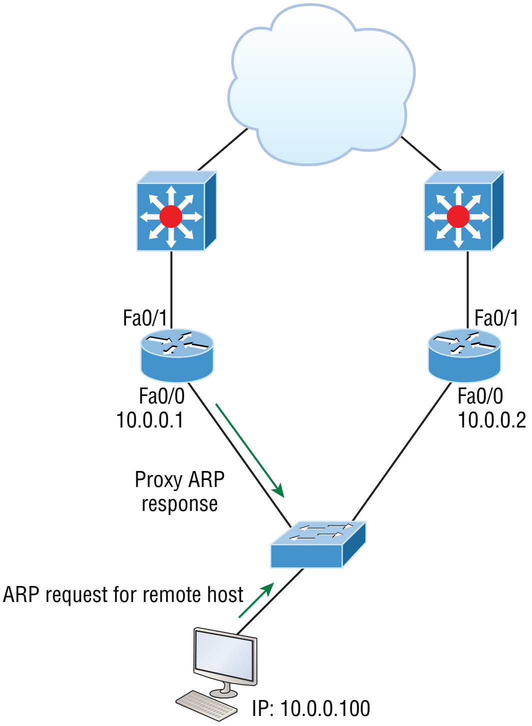


You can see how this happens in Figure 3.2. If a Proxy ARP–enabled router receives an ARP request for an IP address that it knows isn't on the same subnet as the requesting host, it will respond with an ARP reply packet to the host. The router will give its own local MAC address—the MAC address of its interface on the host's subnet—as the destination MAC address for the IP address that the host is seeking to be resolved. After receiving the destination MAC address, the host will then send all the packets to the router, not knowing that what it sees as the destination host is really a router. The router will then forward the packets toward the intended host.

So with Proxy ARP, the host device sends traffic as if the destination device were located on its own network segment. If the router that responded to the ARP request fails, the source host continues to send packets for that destination to the same MAC address. But because they're being sent to a failed router, the packets will be sent to the other router that is also responding to ARP requests for remote hosts.

After the time-out period on the host, the proxy ARP MAC address ages out of the ARP cache. The host can then make a new ARP request for the destination and get the address of another proxy ARP router. Still, keep in mind that the host cannot send packets off of its subnet during the failover time. This isn't exactly a perfect situation, so there has to be a better way, right? Well, there is, and that's precisely where redundancy protocols come to the rescue!

FIGURE 3.2 Proxy ARP

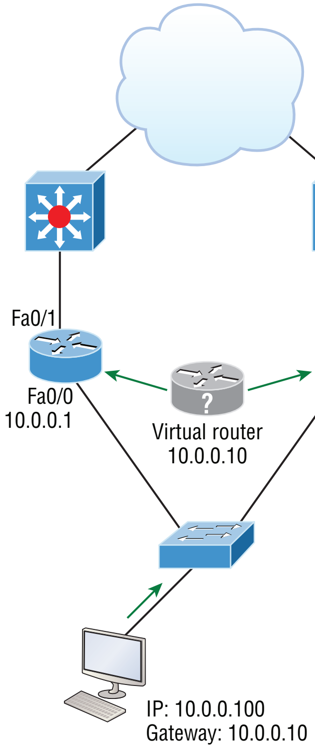


Introducing First Hop Redundancy Protocol (FHRP)

First hop redundancy protocols (FHRPs) work by giving you a way to configure more than one physical router to appear as if they were only a single logical one. This makes client configuration and communication easier because you can simply configure a single default gateway and the host machine can use its standard protocols to communicate. *First hop* is a reference to the default router being the first router, or first router hop, through which a packet must pass.

So how does a redundancy protocol accomplish this? The protocols I'm going to describe to you do this basically by presenting a virtual router to all of the clients. The virtual router has its own IP and MAC addresses. The virtual IP address is the address that's configured on each of the host machines as the default gateway. The virtual MAC address is the address that will be returned when an ARP request is sent by a host. The hosts don't know or care which physical router is actually forwarding the traffic, as you can see in Figure 3.3.

FIGURE 3.3 FHRPs use a virtual router with a virtual IP address and virtual MAC address.



It's the responsibility of the redundancy protocol to decide which physical router will actively forward traffic and which one will be placed on standby in case the active router fails. Even if the active router fails, the transition to the standby router will be transparent to the hosts because the virtual router, which is identified by the virtual IP and MAC addresses, is now used by the standby router. The hosts never change default gateway information, so traffic keeps flowing.

NOTE Fault-tolerant solutions provide continued operation in the event of a device failure, and load-balancing solutions distribute the workload over multiple devices.

Next we'll explore these three important redundancy protocols:

Hot Standby Router Protocol (HSRP) is by far Cisco's favorite protocol ever! Don't buy just one router; buy up to eight routers to provide the same service, and keep seven as backup in case of failure! HSRP is a Cisco proprietary protocol that provides a redundant gateway for hosts on a local subnet, but this isn't a load-balanced solution. HSRP allows you to configure two or more routers into a standby group that shares an IP address and MAC address and provides a default gateway. When the IP and MAC addresses are independent from the routers' physical addresses (on a virtual interface, not tied to a specific interface), they can swap control of an address if the current forwarding and active router fails. But there is actually a way you can sort of achieve load balancing with HSRP—by using multiple VLANs and designating a specific router active for one VLAN, then an alternate router as active for the other VLAN via trunking. This still isn't a true load-balancing solution and it's not nearly as solid as what you can achieve with GLBP!

Virtual Router Redundancy Protocol (VRRP) also provides a redundant—but again, not load-balanced—gateway for hosts on a local subnet. It's an open standard protocol that functions almost identically to HSRP. I'll comb through the fine differences that exist between these protocols later in the chapter.

Gateway Load Balancing Protocol (GLBP) doesn't just stop at providing us with a redundant gateway; it's a true load-balancing solution for routers. GLBP allows a maximum of four routers in each forwarding group. By default, the active router directs the traffic from hosts to each successive router in the group using a round-robin algorithm. The hosts are directed to send their traffic toward a specific router by being given the MAC address of the next router in line to be used.

Hot Standby Router Protocol (HSRP)

Again, HSRP is a Cisco proprietary protocol that can be run on most, but not all, of Cisco's router and multilayer switch models. It defines a standby group, and each standby group that you define includes the following routers:

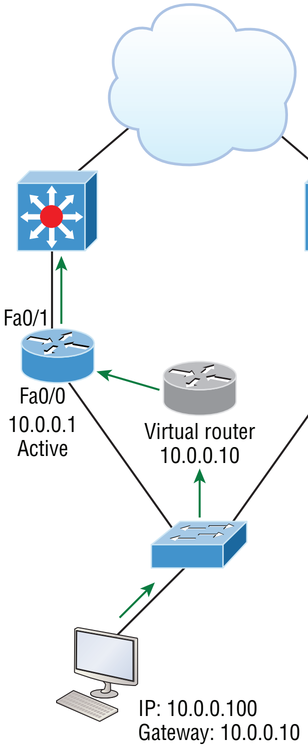
- Active router
- Standby router
- Virtual router
- Any other routers that maybe attached to the subnet

The problem with HSRP is that with it, only one router is active and two or more routers just sit there in standby mode and won't be used unless a failure occurs—not very cost effective or efficient! Figure 3.4 shows how only one router is used at a time in an HSRP group.

The standby group will always have at least two routers participating in it. The primary players in the group are the one active router and one standby router that communicate to each other using multicast Hello messages. The Hello messages provide all of the required communication for the routers. The Hellos contain the information required to accomplish the election that determines the active and standby router positions. They also hold the key to the failover process. If the standby router stops receiving Hello packets from the active router, it then takes over the active router role, as shown in Figure 3.5.

As soon as the active router stops responding to Hellos, the standby router automatically becomes the active router and starts responding to host requests.

FIGURE 3.4 HSRP active and standby routers



Virtual MAC Address

A virtual router in an HSRP group has a virtual IP address and a virtual MAC address. So where does that virtual MAC come from? The virtual IP address isn't that hard to figure out; it just has to be a unique IP address on the same subnet as the hosts defined in the configuration. But MAC addresses are a little different, right? Or are they? The answer is yes—sort of. With HSRP, you create a totally new, made-up MAC address in addition to the IP address.

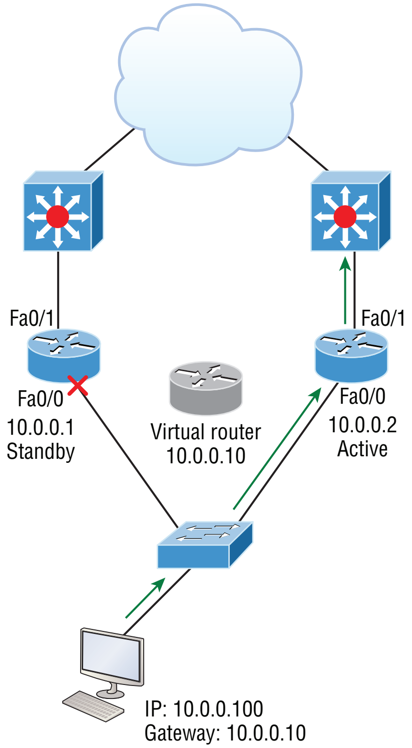
The HSRP MAC address has only one variable piece in it. The first 24 bits still identify the vendor who manufactured the device (the organizationally unique identifier, or OUI). The next 16 bits in the address tell us that the MAC address is a well-known HSRP MAC address. Finally, the last 8 bits of the address are the hexadecimal representation of the HSRP group number.

Let me clarify all this with an example of what an HSRP MAC address would look like:

```
0000.0c07.ac0a
```

- The first 24 bits (0000.0c) are the vendor ID of the address; in the case of HSRP being a Cisco protocol, the ID is assigned to Cisco.
- The next 16 bits (07.ac) are the well-known HSRP ID. This part of the address was assigned by Cisco in the protocol, so it's always easy to recognize that this address is for use with HSRP.
- The last 8 bits (0a) are the only variable bits and represent the HSRP group number that you assign. In this case, the group number is 10 and converted to hexadecimal when placed in the MAC address, where it becomes the 0a that you see.

FIGURE 3.5 Example of HSRP active and standby routers swapping interfaces



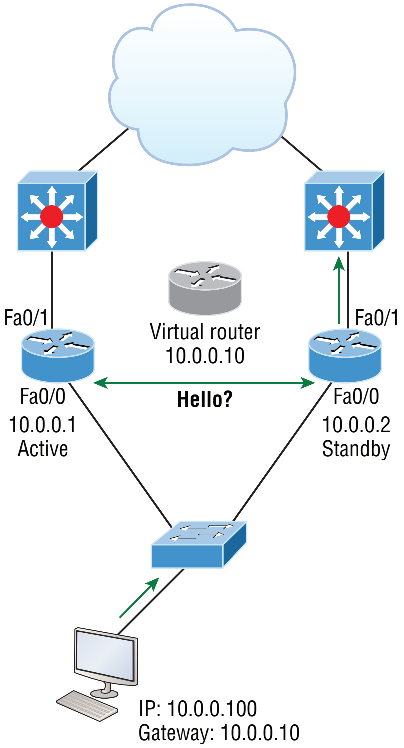
You can see this displayed with every MAC address added to the ARP cache of every router in the HSRP group. There will be the translation from the IP address to the MAC address, as well as the interface on which it's located.

HSRP Timers

Before we get deeper into the roles that each of the routers can have in an HSRP group, I want to define the HSRP timers. The timers are very important to HSRP function because they ensure communication between the routers, and if something goes wrong, they allow the standby router to take over. The HSRP timers include *hello*, *hold*, *active*, and *standby*.

Hello timer The hello timer is the defined interval during which each of the routers send out Hello messages. Their default interval is 3 seconds and they identify the state that each router is in. This is important because the particular state determines the specific role of each router and, as a result, the actions each will take within the group. Figure 3.6 shows the Hello messages being sent and the router using the hello timer to keep the network flowing in case of a failure.

FIGURE 3.6 HSRP Hellos



This timer can be changed and people used to avoid doing so because it was thought that lowering the hello value would place an unnecessary load on the routers. That isn't true with most of the routers today; in fact, you can configure the timers in milliseconds, meaning the failover time can be in milliseconds! Still, keep in mind that increasing the value will make the standby router wait longer before taking over for the active router when it fails or can't communicate.

Hold timer The hold timer specifies the interval the standby router uses to determine whether the active router is offline or out of communication. By default, the hold timer is 10 seconds, roughly three times the default for the hello timer. If one timer is changed for some reason, I recommend using this multiplier to adjust the other timers too. By setting the hold timer at three times the hello timer, you ensure that the standby router doesn't take over the active role every time there's a short break in communication.

Active timer The active timer monitors the state of the active router. The timer resets each time a router in the standby group receives a Hello packet from the active router. This timer expires based on the hold time value that's set in the corresponding field of the HSRP hello message.

Standby timer The standby timer is used to monitor the state of the standby router. The timer resets anytime a router in the standby group receives a Hello packet from the standby router and expires based on the hold time value that's set in the respective Hello packet.

Real World Scenario

Large Enterprise Network Outages with FHRPs

Years ago when HSRP was all the rage, and before VRRP and GLBP, enterprises used hundreds of HSRP groups. With the hello timer set to 3 seconds and a hold time of 10 seconds, these timers worked just fine and we had great redundancy with our core routers.

However, in the last few years, and certainly in the future, 10 seconds is now a lifetime! Some of my customers have been complaining with the failover time and loss of connectivity to their virtual server farm.

So lately I've been changing the timers to well below the defaults. Cisco had changed the timers so you could use sub-second times for failover. Because these are multicast packets, the overhead that is seen on a current high-speed network is almost nothing.

The hello timer is typically set to 200 msec and the hold time is 700 msec. The command is as follows:

```
(config-if)#Standby 1 timers msec 200 msec 700
```

This almost ensures that not even a single packet is lost when there is an outage.

Group Roles

Each of the routers in the standby group has a specific function and role to fulfill. The three main roles are as virtual router, active router, and standby router. Additional routers can also be included in the group.

Virtual router As its name implies, the virtual router is not a physical entity. It really just defines the role that's held by one of the physical routers. The physical router that communicates as the virtual router is the current active router. The virtual router is nothing more than a separate IP address and MAC address to which packets are sent.

Active router The active router is the physical router that receives data sent to the virtual router address and routes it onward to its various destinations. As I mentioned, this router accepts all the data sent to the MAC address of the virtual router in addition to the data that's been sent to its own physical MAC address. The active router processes the data that's being forwarded and will also answer any ARP requests destined for the virtual router's IP address.

Standby router The standby router is the backup to the active router. Its job is to monitor the status of the HSRP group and quickly take over packet-forwarding responsibilities if the active router fails or loses communication. Both the active and standby routers transmit Hello messages to inform all other routers in the group of their role and status.

Other routers An HSRP group can include additional routers, which are members of the group but that don't take the primary roles of either active or standby states. These routers monitor the Hello messages sent by the active and standby routers to ensure that an active and standby router exists for the HSRP group that they belong to. They will forward data that's specifically addressed to their own IP addresses, but they will never forward data addressed to the virtual router unless elected to the active or standby state. These routers send "speak" messages based on the hello timer interval that informs other routers of their position in an election.

Interface Tracking

By now, you probably understand why having a virtual router on a LAN is a great idea. You also know why it's a very good thing that the active router can change dynamically, giving us much needed redundancy on our inside network. But what about the links to the upstream network or the Internet connection off of those HSRP-enabled routers? And how will the inside hosts know if an outside interface goes down or if they are sending packets to an active router that can't route to a remote network? Key questions and HSRP do provide a solution for them called interface tracking.

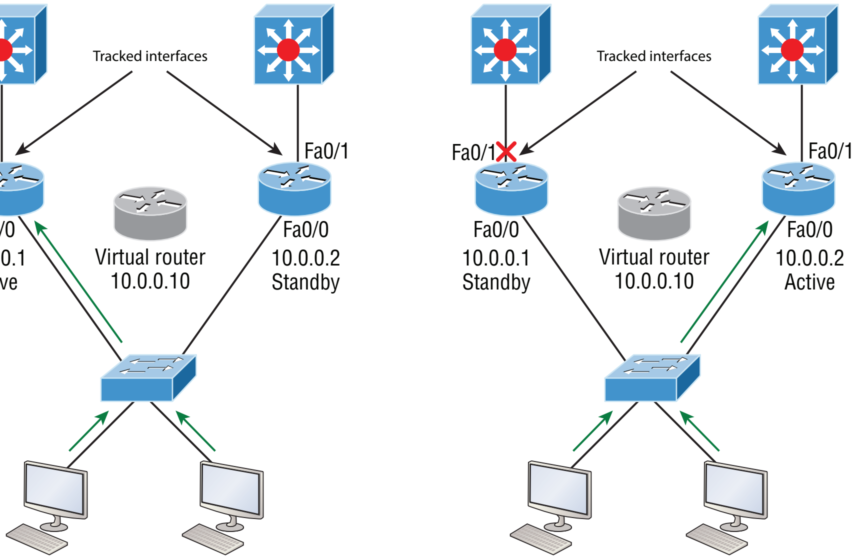
Figure 3.7 shows how HSRP-enabled routers can keep track of the interface status of the outside interfaces and how they can switch the inside active router as needed to keep the inside hosts from losing connectivity upstream.

If the outside link of the active router goes down, the standby router will take over and become the active router. There is a default priority of 100 on routers configured with an HSRP interface, and if you raise this priority (we'll do this in a minute), it means your router has a higher priority to become the active router. The reason I am bringing this up now is because when a tracked interface goes down, it decrements the priority of this router.

Configuring and Verifying HSRP

Configuring and verifying the different FHRPs can be pretty simple, especially regarding the Cisco objectives, but as with most technologies, you can quickly get into advanced configurations and territory with the different FHRPs. The Cisco objectives don't cover much about the configuration of FHRPs, but verification is important, so I'll use a simple configuration on two routers here, starting with HSRP, which I'll also refer to for a simple GLBP configuration and verification later in this chapter. Figure 3.8 shows the network I'll use to demonstrate the FHRPs.

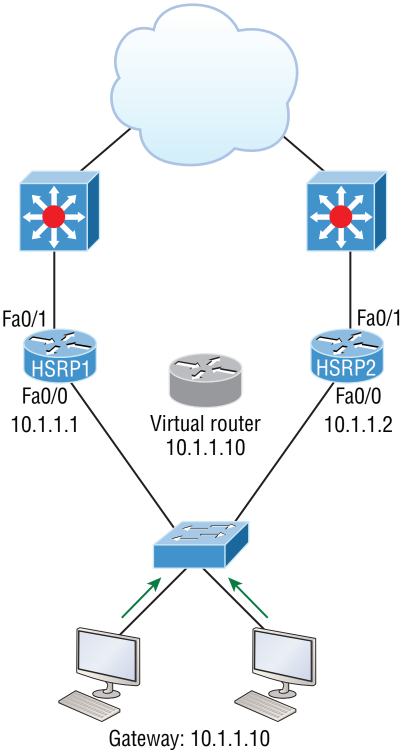
FIGURE 3.7 Interface tracking setup


Tracked interfaces
Fa0/1
Fa0/1
Fa0/0
10.0.0.1
Active
Virtual router
10.0.0.10
Fa0/0
10.0.0.2
Standby
Tracked interfaces
Fa0/1
Fa0/1
Fa0/0
10.0.0.1
Standby
Virtual router
10.0.0.10
Fa0/0
10.0.0.2
Active

This is a simple configuration that you really need only one command for: `standby` *`group`* `ip` *`virtual_ip`*. After using this single mandatory command, I'll name the group and set the interface on router HSRP1 so it wins the election and becomes the active router by default.

```
HSRP1#config t
HSRP1(config)#int fa0/0
HSRP1(config-if)#standby ?
  <0-255>         group number
  authentication  Authentication
  delay           HSRP initialisation delay
  ip              Enable HSRP and set the virtual IP address
  mac-address     Virtual MAC address
  name            Redundancy name string
  preempt         Overthrow lower priority Active routers
  priority        Priority level
  redirect        Configure sending of ICMP Redirect messages with an HSRP
                  virtual IP address as the gateway IP address
  timers          Hello and hold timers
  track           Priority tracking
  use-bia         HSRP uses interface's burned in address
  version         HSRP version

HSRP1(config-if)#standby 1 ip 10.1.1.10
HSRP1(config-if)#standby 1 name HSRP_Test
HSRP1(config-if)#standby 1 priority ?
  <0-255>  Priority value

HSRP1(config-if)#standby 1 priority 110
000047: %HSRP-5-STATECHANGE: FastEthernet0/0 Grp 1 state Speak -> Standby
000048: %HSRP-5-STATECHANGE: FastEthernet0/0 Grp 1 state Standby -> Active110
```

FIGURE 3.8 HSRP configuration and verification


Fa0/1
HSRP1
Fa0/0
10.1.1.1
Fa0/1
HSRP2
Fa0/0
10.1.1.2
Virtual router
10.1.1.10
Gateway: 10.1.1.10

There are quite a few commands available to use in an advanced setting with the `standby` command, but we'll stick with the simple commands that follow the Cisco objectives. First, I numbered the group (1), which must be the same on all routers sharing HSRP duties; then I added the virtual IP address shared by all routers in the HSRP group. Optionally, I named the group and then set the priority of HSRP1 to 110, and I left HSRP2 to a default of 100. The router with the highest priority will win the election to become the active router. Let's configure the HSRP2 router now:

```
HSRP2#config t
HSRP2(config)#int fa0/0
HSRP2(config-if)#standby 1 ip 10.1.1.10
HSRP2(config-if)#standby 1 name HSRP_Test
*Jun 23 21:40:10.699:%HSRP-5-STATECHANGE:FastEthernet0/0 Grp 1 state
Speak -> Standby
```

I really only needed the first command—naming it was for administrative purposes only. Notice that the link came up and HSRP2 became the standby router because it had the lower priority of 100 (the default). Make a note that this priority comes into play only if both routers were to come up at the same time. This means that HSRP2 would be the active router, regardless of the priority, if it comes up first.

Let's take a look at the configurations with the `show standby` and `show standby brief` commands:

```
HSRP1(config-if)#do show standby
FastEthernet0/0 - Group 1
  State is Active
    2 state changes, last state change 00:03:40
  Virtual IP address is 10.1.1.10
  Active virtual MAC address is 0000.0c07.ac01
    Local virtual MAC address is 0000.0c07.ac01 (v1 default)
  Hello time 3 sec, hold time 10 sec
    Next hello sent in 1.076 secs
  Preemption disabled
  Active router is local
  Standby router is 10.1.1.2, priority 100 (expires in 7.448 sec)
  Priority 110 (configured 110)
  IP redundancy name is "HSRP_Test" (cfgd)


HSRP1(config-if)#do show standby brief
                     P indicates configured to preempt.
                     |
Interface   Grp Prio P State    Active          Standby         Virtual IP
Fa0/0       1   110    Active   local           10.1.1.2        10.1.1.10
```

Notice the group number in each output—it's a key troubleshooting spot! Each router must be configured in the same group or they won't work. Also, you can see the virtual MAC and configured virtual IP address, as well as the hello time of 3 seconds. The standby and virtual IP addresses are also displayed.

HSRP2's output tells us that it's in standby mode:

```
HSRP2(config-if)#do show standby brief
                     P indicates configured to preempt.
                     |
Interface   Grp Prio P State    Active          Standby         Virtual IP
Fa0/0       1   100    Standby  10.1.1.1        local           10.1.1.10
HRSP2(config-if)#
```

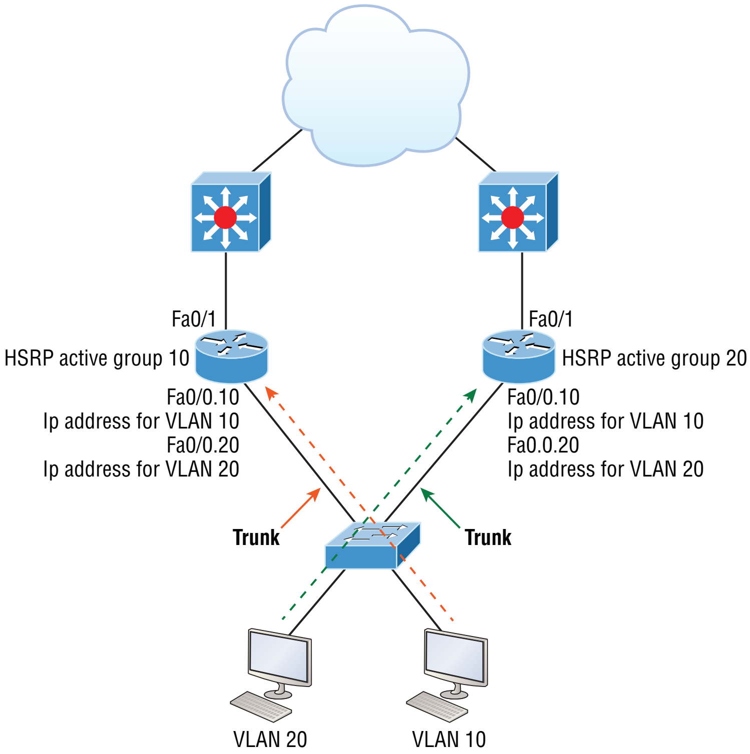
There is one other command that I want to cover. If you're studying and want to understand HSRP, you should learn to use this `debug` command and have your active and standby routers move. You'll really get to see what is going on.

```
HSRP2#debug standby
*Sep 15 00:07:32.344:HSRP:Fa0/0 Interface UP
*Sep 15 00:07:32.344:HSRP:Fa0/0 Initialize swsb, Intf state Up
*Sep 15 00:07:32.344:HSRP:Fa0/0 Starting minimum intf delay (1 secs)
*Sep 15 00:07:32.344:HSRP:Fa0/0 Grp 1 Set virtual MAC 0000.0c07.ac01
type: v1 default
*Sep 15 00:07:32.344:HSRP:Fa0/0 MAC hash entry 0000.0c07.ac01, Added
Fa0/0 Grp 1 to list
*Sep 15 00:07:32.348:HSRP:Fa0/0 Added 10.1.1.10 to hash table
*Sep 15 00:07:32.348:HSRP:Fa0/0 Grp 1 Has mac changed? cur 0000.0c07.ac01
new 0000.0c07.ac01
*Sep 15 00:07:32.348:HSRP:Fa0/0 Grp 1 Disabled -> Init
*Sep 15 00:07:32.348:HSRP:Fa0/0 Grp 1 Redundancy "hsrp-Fa0/0-1" state Disabled ->
Init
*Sep 15 00:07:32.348:HSRP:Fa0/0 IP Redundancy "hsrp-Fa0/0-1" added
*Sep 15 00:07:32.348:HSRP:Fa0/0 IP Redundancy "hsrp-Fa0/0-1" update,
Disabled -> Init
*Sep 15 00:07:33.352:HSRP:Fa0/0 Intf min delay expired
*Sep 15 00:07:39.936:HSRP:Fa0/0 Grp 1 MAC addr update Delete from SMF  0000.0c07.
ac01
*Sep 15 00:07:39.936:HSRP:Fa0/0 Grp 1 MAC addr update Delete from SMF  0000.0c07.
ac01
*Sep 15 00:07:39.940:HSRP:Fa0/0 ARP reload
```

HSRP Load Balancing

As you know, HSRP doesn't really perform true load balancing, but it can be configured to use more than one router at a time for use with different VLANs. This is different from the true load balancing that's possible with GLBP, which I'll demonstrate in a minute, but HSRP still performs a load-balancing act of sorts. Figure 3.9 shows how load balancing would look with HSRP.

FIGURE 3.9 HSRP load balancing per VLAN


Fa0/1
HSRP active group 10
Fa0/0.10
Ip address for VLAN 10
Fa0/0.20
Ip address for VLAN 20
Fa0/1
HSRP active group 20
Fa0/0.10
Ip address for VLAN 10
Fa0.0.20
Ip address for VLAN 20
Trunk
Trunk
VLAN 20
VLAN 10

How can you get two HSRP routers active at the same time? Well for the same subnet with this simple configuration you can't, but by trunking the links to each router, they'll run and be configured with a "router on a stick" (ROAS) configuration. This means that each router can be the default gateway for different VLANs, but you still can have only one active router per VLAN. Typically, in a more advanced setting you won't use HSRP for load balancing; you'll use GLBP, but you can do load-sharing with HSRP, and that is the topic of an objective, so we'll remember that, right? It comes in handy because it prevents situations where a single point of failure causes traffic interruptions. This HSRP feature improves network resilience by allowing for load-balancing and redundancy capabilities between subnets and VLANs.

Virtual Router Redundancy Protocol

Like HSRP, Virtual Router Redundancy Protocol (VRRP) allows a group of routers to form a single virtual router. In an HSRP or VRRP group, one router is elected to handle all requests sent to the virtual IP address. With HSRP, this is the active router. An HSRP group has one active router, at least one standby router, and many listening routers. A VRRP group has one master router and one or more backup routers and is the open standard implementation of HSRP.

Comparing VRRP and HSRP

The LAN workstations are configured with the address of the virtual router as their default gateway, just as they are with HSRP, but VRRP differs from HSRP in these important ways:

- VRRP is an IEEE standard (RFC 2338) for router redundancy; HSRP is a Cisco proprietary protocol.
- The virtual router that represents a group of routers is known as a VRRP group.
- The active router is referred to as the master virtual router.
- The master virtual router may have the same IP address as the virtual router group.
- Multiple routers can function as backup routers.
- VRRP is supported on Ethernet, Fast Ethernet, and Gigabit Ethernet interfaces, as well as on Multiprotocol Label Switching (MPLS) virtual private networks (VPNs) and VLANs.

VRRP Redundancy Characteristics

VRRP has some unique features:

- VRRP provides redundancy for the real IP address of a router or for a virtual IP address shared among the VRRP group members.
- If a real IP address is used, the router with that address becomes the master.
- If a virtual IP address is used, the master is the router with the highest priority.
- A VRRP group has one master router and one or more backup routers.
- The master router uses VRRP messages to inform group members of its status.
- VRRP allows load sharing across more than one virtual router.

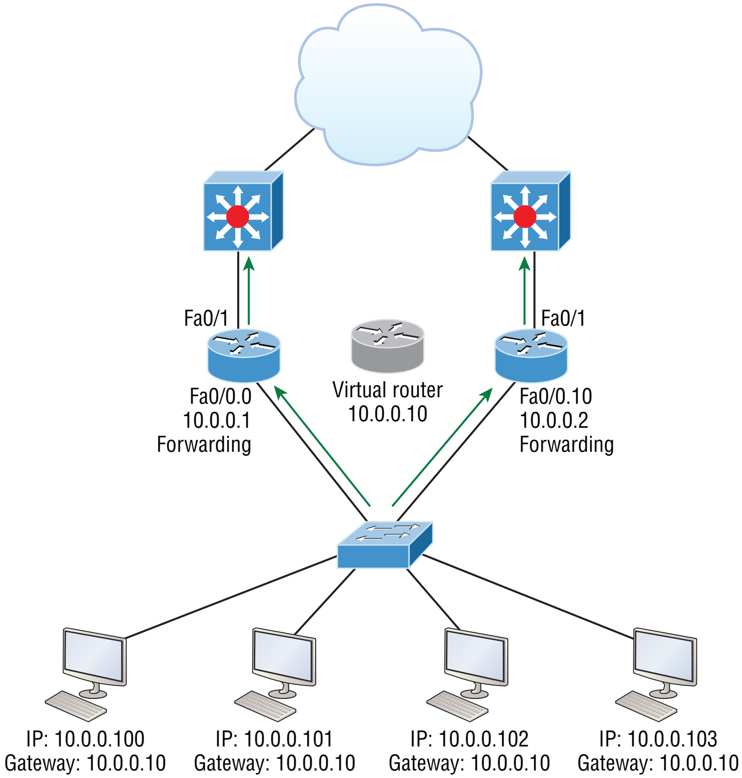
Gateway Load Balancing Protocol

Although HSRP and VRRP provide gateway resiliency with per-subnet load balancing, the upstream bandwidth of the standby members of the redundancy group isn't used while the devices are in standby mode.

Only the active routers in HSRP and VRRP groups forward traffic for the virtual MAC. Resources associated with the standby router are not fully utilized. Some load-balancing can be accomplished with these protocols through the creation of multiple groups and through the assignment of multiple default gateways, but be warned—these configurations create an administrative burden and are inefficient for today's networks!

Cisco designed a proprietary load-balancing protocol, Gateway Load Balancing Protocol (GLBP), to allow automatic selection and simultaneous use of multiple available gateways, as well as permit automatic failover between those gateways. GLBP takes an active/active approach on a per-subnet basis to support first-hop (default router) traffic when implemented with two routers on the same LAN. Multiple routers share the load of frames that, from a client perspective, are sent to a single default gateway address, as shown in Figure 3.10.

FIGURE 3.10 Gateway Load Balancing Protocol (GLBP)


Fa0/1
Fa0/1
Fa0/0.0
10.0.0.1
Forwarding
Virtual router
10.0.0.10
Fa0/0.10
10.0.0.2
Forwarding
IP: 10.0.0.100
Gateway: 10.0.0.10
IP: 10.0.0.101
Gateway: 10.0.0.10
IP: 10.0.0.102
Gateway: 10.0.0.10
IP: 10.0.0.103
Gateway: 10.0.0.10

With GLBP, resources can be fully utilized without the administrative hassle of configuring multiple groups and managing multiple default gateway configurations as required when working with HSRP and VRRP.

GLBP Functions

GLBP essentially provides clients with the following:

- An active virtual gateway (AVG)
- An active virtual forwarder (AVF)

It also allows members of the group to communicate with each other through Hello messages sent every 3 seconds to the multicast address 224.0.0.102, User Datagram Protocol (UDP) port 3222.

GLBP AVG Members of a GLBP group elect one gateway to be the AVG for that group. Other group members provide backup for the AVG in the event that the AVG becomes unavailable. The AVG assigns a different virtual MAC address to each member of the GLBP group.

GLBP AVF Each gateway assumes responsibility for forwarding packets that are sent to the virtual MAC address assigned to that gateway by the AVG. These gateways are known as AVFs for their virtual MAC address.

GLBP Features

GLBP provides upstream load-sharing by utilizing the redundant uplinks simultaneously. It uses link capacity efficiently, thus providing solid peak-load traffic coverage. By making use of multiple available paths upstream from the routers or layer 3 switches running GLBP, you can also reduce output queues. GLBP supports the following features:

Load sharing You can configure GLBP so that traffic from LAN clients is shared by multiple routers. As the name suggests, load sharing distributes the traffic load more evenly among available routers.

Multiple virtual routers GLBP supports up to 1,024 virtual routers as GLBP groups on each router's physical interface and up to four virtual forwarders per group.

Preemption According to the dictionary, *preempt* means "to replace with something considered to be of greater value or priority." The redundancy scheme of GLBP allows us to preempt an AVG with a higher-priority backup virtual gateway that has become available. Forwarder preemption works in a similar way, except that it's based upon weighting instead of priority and is enabled by default. One router can take over another router only during an election, and preemption is the only way to force an election when a device hasn't gone down.

Efficient resource utilization GLBP makes it possible for any router in a group to serve as a backup, which eliminates the need for a dedicated backup router because all available routers can support network traffic.

Remember that only a single path is used with HSRP or VRRP, while other resources are idle unless you've got multiple groups and gateways configured. This means that a single path can be subjected to higher output queue rates during peak times, leading to lower performance caused by higher jitter rates. The good news is that we can mitigate the impact of jitter with GLBP because with it, more upstream bandwidth is available and additional upstream paths are used.

Further, GLBP permits automatic selection and simultaneous use of all available gateways in the group. The members of a GLBP group elect one gateway to be the AVG for that group, and other members of the group provide backup for the AVG if it becomes unavailable. The AVG assigns a virtual MAC address to each member of the GLBP group. All routers become AVFs for frames addressed to that specific virtual MAC address. And as clients send ARP requests for the address of the default gateway, the AVG sends these virtual MAC addresses in the ARP replies. Don't forget that a GLBP group can have up to four group members.

GLBP automatically manages the virtual MAC address assignment, determines who handles the forwarding, and ensures that each host has a reliable forwarding path if failures to gateways or tracked interfaces occur. Also, when failures do occur, the load-balancing ratio is adjusted among the remaining AVFs so that resources are used in the most efficient way—nice!

GLBP Per-host Traffic Balancing

These two steps will really help clarify how GLBP balances traffic using the round-robin algorithm:

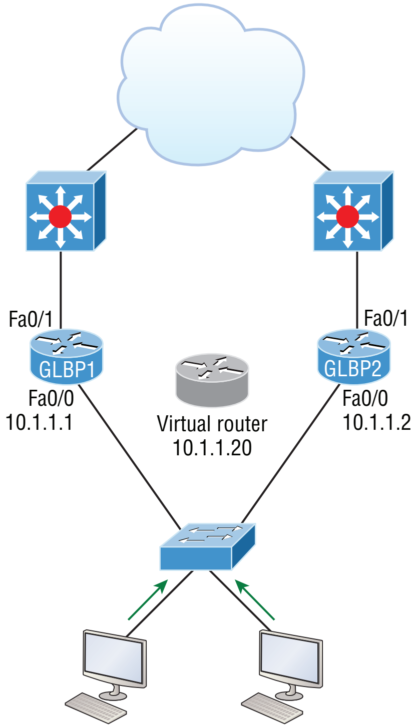
1. When a client sends an ARP message for the gateway IP address, the AVG returns the virtual MAC address of one of the AVFs.
2. When a second client sends an ARP message, the AVG returns the next virtual MAC address from the list.

So having resolved a different MAC address for the default gateway, each client will send its routed traffic to separate routers even though they both have the same default gateway address configured. Remember that each GLBP router will be the designated AVF for the specific virtual MAC address that's been assigned to it.

Configuring GLBP

Let's go ahead and configure GLBP on our little internetwork now, as shown in Figure 3.11. I'll use the same minimal configuration, with GLBP1 getting a higher priority.

FIGURE 3.11 Configuring GLBP


Fa0/1
GLBP1
Fa0/0
10.1.1.1
Virtual router
10.1.1.20
Fa0/1
GLBP2
Fa0/0
10.1.1.2

```
GLBP1(config)#int fa0/0
GLBP1(config-if)#glbp 1 ip 10.1.1.20
GLBP1(config-if)#glbp 1 name GLBP_Test
GLBP1(config-if)#GLBP 1 priority 110
GLBP1(config-if)#
000050:%GLBP-6-STATECHANGE:FastEthernet0/0 Grp 1 state Standby -> Active
000051:%GLBP-6-FWDSTATECHANGE:FastEthernet0/0 Grp 1 Fwd 1 state Listen -> Active
```

Next, I'll configure GLBP2:

```
GLBP2(config)#int fa0/0
GLBP2(config-if)#glbp 1 ip 10.1.1.20
GLBP2(config-if)#glbp 1 name GLBP_Test
GLBP2(config-if)#
*Jun 23 21:49:16.059:%GLBP-6-FWDSTATECHANGE:FastEthernet0/0 Grp 1 Fwd 2
state Listen -> Active
```

Okay—this is pretty much the same configuration as HSRP, except that I used `glbp` instead of the `standby` command. Let's see which router is active now:

```
GLBP1(config-if)#do sh glbp
FastEthernet0/0 - Group 1
  State is Active
    2 state changes, last state change 00:02:29
  Virtual IP address is 10.1.1.20
  Hello time 3 sec, hold time 10 sec
    Next hello sent in 0.140 secs
  Redirect time 600 sec, forwarder time-out 14400 sec
  Preemption disabled
  Active is local
  Standby is 10.1.1.2, priority 100 (expires in 8.444 sec)
  Priority 110 (configured)
  Weighting 100 (default 100), thresholds: lower 1, upper 100
  Load balancing: round-robin
  IP redundancy name is "GLBP_Test"
  Group members:
    001a.2fe7.4398 (10.1.1.1) local
    001a.6ca1.1f48 (10.1.1.2)
  There are 2 forwarders (1 active)
  Forwarder 1
    State is Active
      1 state change, last state change 00:02:19
    MAC address is 0007.b400.0101 (default)
    Owner ID is 001a.2fe7.4398
    Redirection enabled
    Preemption enabled, min delay 30 sec
    Active is local, weighting 100
  Forwarder 2
    State is Listen
    MAC address is 0007.b400.0102 (learnt)
    Owner ID is 001a.6ca1.1f48
    Redirection enabled, 599.788 sec remaining (maximum 600 sec)
    Time to live: 14399.788 sec (maximum 14400 sec)
    Preemption enabled, min delay 30 sec
    Active is 10.1.1.2 (primary), weighting 100 (expires in 9.788 sec)
GLBP1(config-if)#do sh glbp brief
Interface   Grp  Fwd Pri State    Address         Active router   Standby route
Fa0/0       1    -   110 Active   10.1.1.20       local           10.1.1.2
```

```
Fa0/0       1    1   7   Active   0007.b400.0101  local          -
Fa0/0       1    2   7   Listen   0007.b400.0102  10.1.1.2       -
```

Wow—the `show glbp` command gave us a lot more output than the `show standby` command did because there's more to GLBP than there is to HSRP! For example, you can see that there are two forwarders listed under the `Fwd` heading.

And there are three lines associated with the `show glbp brief` command. The first is the AVG, the dash (-) that shows up under the `Fwd` column. Don't get confused about the active and listen states in the output because they just indicate which router will respond to ARP requests for the virtual IP address. This example highlights GLBP1 as the active router. Make sure to note the two MAC addresses under the `Address` column because they reveal the virtual MAC addresses used by the routers.

Syslog

Reading system messages from a switch's or router's internal buffer is the most popular and efficient method of seeing what's going on with your network at a particular time. But the best way is to log messages to a *syslog* server, which stores messages from you and can even time-stamp and sequence them for you, and it's easy to set up and configure!

Syslog allows you to display, sort, and even search messages, all of which makes it a really great troubleshooting tool. The search feature is especially powerful because you can use keywords and even severity levels. Plus, the server can email admins based on the severity level of the message.

Network devices can be configured to generate a syslog message and forward it to various destinations. These four examples are popular ways to gather messages from Cisco devices:

- Logging buffer (on by default)
- Console line (on by default)
- Terminal lines (using the `terminal monitor` command)
- Syslog server

As you already know, all system messages and debug output generated by the IOS go out only the console port by default and are also logged in buffers in RAM. And you also know that Cisco routers aren't exactly shy about sending messages! To send message to the VTY lines, use the `terminal monitor` command. We'll also add a small configuration needed for syslog, which I'll show you soon in the configuration section.

Okay, so by default, we'd see something like this on our console line:

```
*Oct 21 17:33:50.565:%LINK-5-CHANGED:Interface FastEthernet0/0, changed
state to administratively down
*Oct 21 17:33:51.565:%LINEPROTO-5-UPDOWN:Line protocol on Interface
FastEthernet0/0, changed state to down
```

And the Cisco router would send a general version of the message to the syslog server that would be formatted into something like this:

```
Seq no:timestamp: %facility-severity-MNEMONIC:description
```

The system message format can be broken down in this way:

seq no This stamp logs messages with a sequence number, but not by default. If you want this output, you've got to configure it.

Timestamp Data and time of the message or event, which again will show up only if configured.

Facility The facility to which the message refers.

Severity A single-digit code from 0 to 7 that indicates the severity of the message.

MNEMONIC Text string that uniquely describes the message.

Description Text string containing detailed information about the event being reported.

The severity levels, from the most severe level to the least severe, are explained in Table 3.1. Informational is the default and will result in all messages being sent to the buffers and console.

TABLE 3.1 Severity levels

Severity Level	Explanation
Emergency (severity 0)	System is unusable.
Alert (severity 1)	Immediate action is needed.
Critical (severity 2)	Critical condition.
Error (severity 3)	Error condition.
Warning (severity 4)	Warning condition.
Notification (severity 5)	Normal but significant condition.
Information (severity 6)	Normal information message.
Debugging (severity 7)	Debugging message.

TIP

If you are studying for your Cisco exam, you need to memorize Table 3.1.

Understand that only emergency-level messages will be displayed if you've configured severity level 0. But if, for example, you opt for level 4 instead, level 0 through 4 will be displayed, giving you emergency, alert, critical, error, and warning messages too. Level 7 is the highest-level security option and displays everything, but be warned that going with it could have a serious impact on the performance of your device. So always use debugging commands carefully with an eye on the messages you really need to meet your specific business requirements!

Configuring and Verifying Syslog

As I said, Cisco devices send all log messages of the severity level you've chosen to the console. They'll also go to the buffer, and both happen by default. Because of this, it's good to know that you can disable and enable these features with the following commands:

```
Router(config)#logging ?
  Hostname or A.B.C.D  IP address of the logging host
  buffered             Set buffered logging parameters
  buginf               Enable buginf logging for debugging
  cns-events           Set CNS Event logging level
  console              Set console logging parameters
  count                Count every log message and timestamp last occurrence
  esm                  Set ESM filter restrictions
  exception            Limit size of exception flush output
  facility             Facility parameter for syslog messages
  filter               Specify logging filter
  history              Configure syslog history table
  host                 Set syslog server IP address and parameters
  monitor              Set terminal line (monitor) logging parameters
  on                   Enable logging to all enabled destinations
  origin-id            Add origin ID to syslog messages
  queue-limit          Set logger message queue size
  rate-limit           Set messages per second limit
  reload               Set reload logging level
  server-arp           Enable sending ARP requests for syslog servers when
                       first configured
  source-interface     Specify interface for source address in logging
                       transactions
  trap                 Set syslog server logging level
  userinfo             Enable logging of user info on privileged mode enabling


Router(config)#logging console
Router(config)#logging buffered
```

Wow—as you can see in this output, there are plenty of options you can use with the logging command! The preceding configuration enabled the console and buffer to receive all log message of all severities, and don't forget that this is the default setting for all Cisco IOS devices. If you want to disable the defaults, use the following commands:

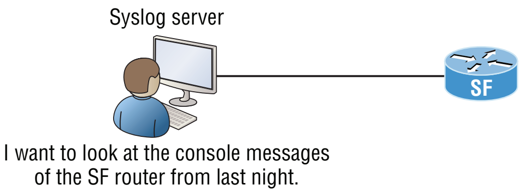
```
Router(config)#no logging console
Router(config)#no logging buffered
```

I like leaving the console and buffers commands on in order to receive the logging info, but that's up to you. You can see the buffers with the `show logging` command here:

```
Router#sh logging
Syslog logging: enabled (11 messages dropped, 1 messages rate-limited,
                0 flushes, 0 overruns, xml disabled, filtering disabled)
    Console logging: level debugging, 29 messages logged, xml disabled,
                     filtering disabled
    Monitor logging: level debugging, 0 messages logged, xml disabled,
                     filtering disabled
    Buffer logging: level debugging, 1 messages logged, xml disabled,
                    filtering disabled
    Logging Exception size (4096 bytes)
    Count and timestamp logging messages: disabled
No active filter modules.


    Trap logging: level informational, 33 message lines logged


Log Buffer (4096 bytes):
*Jun 21 23:09:37.822: %SYS-5-CONFIG_I: Configured from console by console
Router#
```

Notice that the default trap (message from device to NMS) level is informational, but you can change this too. And now that you've seen the default system message format on a Cisco device, I want to show you how you can also control the format of your messages via sequence numbers and time stamps, which aren't enabled by default. We'll begin with a basic, very simple example of how to configure a device to send messages to a syslog server, demonstrated in Figure 3.12.

FIGURE 3.12 Messages sent to a syslog server


Syslog server
SF
I want to look at the console messages of the SF router from last night.

A syslog server saves copies of console messages and can time-stamp them for viewing at a later time. This is actually pretty easy to configure and here's how doing that would look on the SF router:

```
SF(config)#logging 172.16.10.1
SF(config)#service timestamps log datetime msec
```

This is awesome—now all the console messages will be stored in one location to be viewed at your convenience! I typically use the `logging host` *`ip_address`* command, but `logging` *`IP_address`* command without the `host` keyword gets the same result.

I want to point out that even though I had the messages time-stamped in the configuration associated with Figure 3.12, the command `service timestamps log datetime msec` doesn't mean that I'll know the messages' exact time if I'm using default clock sources. To make sure all devices are synchronized with the same time information, make sure you use an NTP server.

We can limit the amount of messages sent to the syslog server, based on severity with the following command:

```
SF(config)#logging trap ?
  <0-7>          Logging severity level
  alerts         Immediate action needed           (severity=1)
  critical       Critical conditions               (severity=2)
  debugging      Debugging messages                (severity=7)
  emergencies    System is unusable                (severity=0)
  errors         Error conditions                  (severity=3)
  informational  Informational messages            (severity=6)
  notifications  Normal but significant conditions (severity=5)
  warnings       Warning conditions                (severity=4)
  <cr>
SF(config)#logging trap warnings
```

Notice that we can use either the number or the actual severity level name—and they are in alphabetical order, not severity order, which makes it even harder to memorize the order! (Thanks, Cisco!) Since I went with severity level 4, I'll receive messages for levels 0 through 4. Now let's configure the router to use sequence numbers:

```
SF(config)#no service timestamps
SF(config)#service sequence-numbers
SF(config)#^Z
000038: %SYS-5-CONFIG_I: Configured from console by console
```

When you exit configuration mode, the router will send a message like the one shown in the preceding code lines. Without the time stamps enabled, we'll no longer see a time and date, but we will see a sequence number.

So we now have the following:

- Sequence number: 000038
- Facility: %SYS
- Severity level: 5
- MNEMONIC: CONFIG_I
- Description: Configured from console by console

I want to stress that of all of these, the security level is what you need to pay attention to the most for the Cisco exams as well as for a means to control the amount of messages sent to the syslog server!

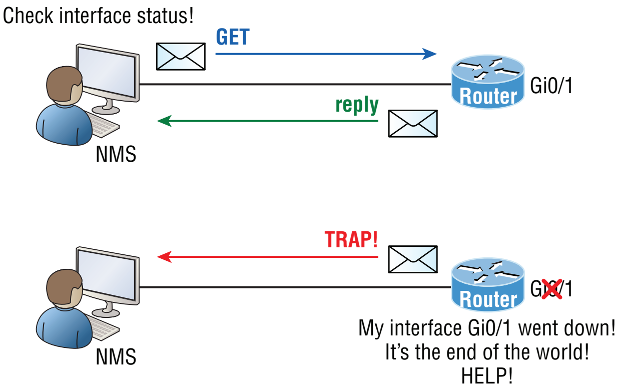
SNMP

Although *Simple Network Management Protocol (SNMP)* certainly isn't the oldest protocol ever, it's still pretty old, considering it was created way back in 1988 (RFC 1065)!

SNMP is an Application layer protocol that provides a message format for agents on a variety of devices to communicate with network management stations (NMSs)—for example, Cisco Prime or HP Openview. These agents send messages to the NMS station which then either reads or writes information in the database stored on the NMS that's called a Management Information Base (MIB).

The NMS periodically queries or polls the SNMP agent on a device to gather and analyze statistics via GET messages. End devices running SNMP agents would send an SNMP trap to the NMS if a problem occurs. This is demonstrated in Figure 3.13.

FIGURE 3.13 SNMP GET and TRAP messages



Admins can also use SNMP to provide some configurations to agents as well, called SET messages. In addition to polling to obtain statistics, SNMP can be used for analyzing information and compiling the results in a report or even a graph. Thresholds can be used to trigger a notification process when exceeded. Graphing tools are used to monitor the CPU statistics of Cisco devices like a core router. The CPU should be monitored continuously and the NMS can graph the statistics. Notification will be sent when any threshold you've set has been exceeded.

SNMP has three versions, with version 1 being rarely, if ever implemented today. Here's a summary of these three versions:

SNMPv1 Supports plaintext authentication with community strings and uses only UDP.

SNMPv2c Supports plaintext authentication with MD5 or SHA with no encryption but provides GET BULK, which is a way to gather many types of information at once and minimize the number of GET requests. It offers a more detailed error message reporting method, but it's not more secure than v1. It uses UDP even though it can be configured to use TCP.

SNMPv3 Supports strong authentication with MD5 or SHA, providing confidentiality (encryption) and data integrity of messages via DES or DES-256 encryption between agents and managers. GET BULK is a supported feature of SNMPv3, and this version also uses TCP.

Management Information Base (MIB)

With so many kinds of devices and so much data that can be accessed, there needed to be a standard way to organize this plethora of data, so MIB to the rescue! A *management information base (MIB)* is a collection of information that's organized hierarchically and can be accessed by protocols like SNMP. RFCs define some common public variables, but most organizations define their own private branches along with basic SNMP standards. Organizational IDs (OIDs) are laid out as a tree with different levels assigned by different organizations, with top-level MIB OIDs belonging to various standards organizations.

Vendors assign private branches in their own products. Let's take a look at Cisco's OIDs, which are described in words or numbers to locate a particular variable in the tree, as shown in Figure 3.14.

Luckily, you don't need to memorize the OIDs in Figure 3.14 for the Cisco exams!

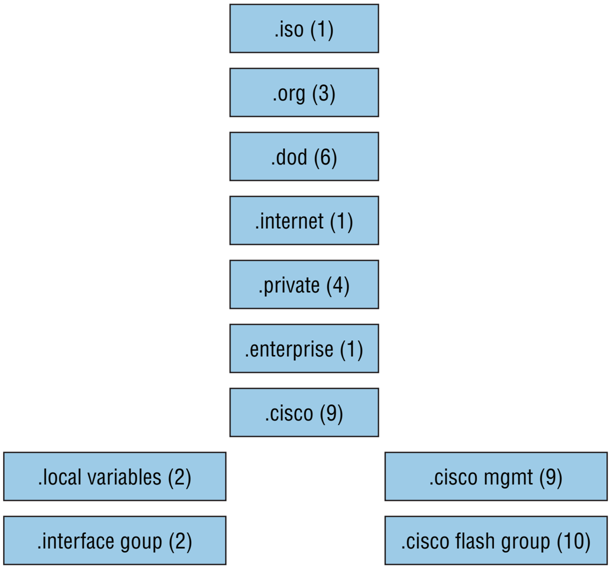
I'll use CPU as an example of a key thing to check at least every 5 minutes. We'll examine output from an SNMP application. It's called snmpget and it comes from an NMS station.

Here's the command from an NMS prompt on a Linux box running the SNMP application:

```
[14:11][admin@nms]$~snmpget -v2c -c community 192.168.10.12
.1.3.6.1.4.1.9.2.1.58.0
SNMPv2-SMI::enterprises.9.2.1.58.0=INTEGER: 19
```

You must specify the version, the correct community string, the IP address of the network device you're querying, plus the OID number. The community string will authenticate your access to the MIB database; and so the NMS can access the switch, the community string definition on the NMS must match at least one of the three community string definitions on the network devices.

FIGURE 3.14 Cisco's MIB OIDs


.iso (1)
.org (3)
.dod (6)
.internet (1)
.private (4)
.enterprise (1)
.cisco (9)
.local variables (2)
.interface goup (2)
.cisco mgmt (9)
.cisco flash group (10)

Configuring SNMP

Configuring SNMP is a pretty straightforward process for which you only need a few commands. These four steps are all you need to run through to configure a Cisco device for SNMP access:

1. Enable SNMP read-write access to the router.
2. Configure SNMP contact information.
3. Configure SNMP location.
4. Configure an ACL to restrict SNMP access to the NMS hosts.

The only required configuration is the community string because the other three are optional. Here's an example of a typical SNMP router configuration:

```
Router(config)#snmp-server ?
  chassis-id        String to uniquely identify this chassis
  community         Enable SNMP; set community string and access privs
  contact           Text for mib object sysContact
  context           Create/Delete a context apart from default
  drop              Silently drop SNMP packets
  enable            Enable SNMP Traps or Informs
  engineID          Configure a local or remote SNMPv3 engineID
  group             Define a User Security Model group
  host              Specify hosts to receive SNMP notifications
  ifindex           Enable ifindex persistence
  inform            Configure SNMP Informs options
  location          Text for mib object sysLocation
  manager           Modify SNMP manager parameters
  packetsize        Largest SNMP packet size
  queue-length      Message queue length for each TRAP host
  source-interface  Assign an source interface
  system-shutdown   Enable use of the SNMP reload command
  tftp-server-list  Limit TFTP servers used via SNMP
  trap              SNMP trap options
  trap-source       Assign an interface for the source address of all traps
  trap-timeout      Set timeout for TRAP message retransmissions
  user              Define a user who can access the SNMP engine
  view              Define an SNMP MIB view

Router(config)#snmp-server community ?
  WORD  SNMP community string

Router(config)#snmp-server community Todd ?
  <1-99>       Std IP accesslist allowing access with this community string
  <1300-1999>  Expanded IP accesslist allowing access with this community
               string
  WORD         Access-list name
  ipv6         Specify IPv6 Named Access-List
  ro           Read-only access with this community string
  rw           Read-write access with this community string
  view         Restrict this community to a named MIB view
  <cr>

Router(config)#snmp-server community Todd rw
Router(config)#snmp-server location Boulder
Router(config)#snmp-server contact Todd Lammle
Router(config)#ip access-list standard Protect_NMS_Station
Router(config-std-nacl)#permit host 192.168.10.254
```

Entering the `snmp-server` command enables SNMPv1 on the Cisco device.

You can enter the ACL directly in the SNMP configuration to provide security, using either a number or a name. Here is an example:

```
Router(config)#snmp-server community Todd Protect_NMS_Station rw
```

Notice that even though there's a boatload of configuration options under SNMP, you only really need to work with a few of them to configure a basic SNMP trap setup on a router. First, I chose the community name of `Todd` with RW access (read-write), which means the NMS will be able to retrieve and modify MIB objects from the router. Location and contact information comes in really handy for troubleshooting the configuration. Make sure you understand that the ACL protects the NMS from access, not the devices with the agents!

Let's define the SNMP read and write options.

Read-only Gives authorized management stations read-access to all objects in the MIB except the community strings and doesn't allow write-access

Read-write Gives authorized management stations read-and write-access to all objects in the MIB but doesn't allow access to the community strings

There are still more ways to gather information from Cisco devices, and next, we'll explore a Cisco proprietary method of gathering statistics on internetwork devices.

Visit ccna .gg/ch3/a for a companion MicroNugget from CBT Nuggets.

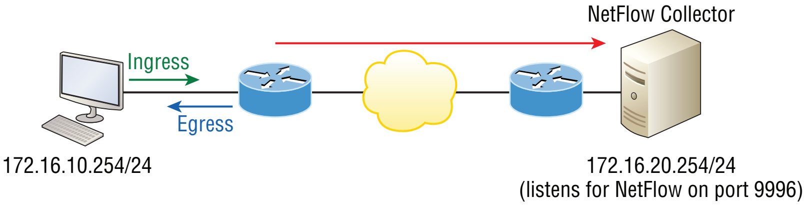
NetFlow

SNMP can be a powerful tool to help you manage and troubleshoot your network, but Cisco knew it would be very helpful for engineers to be able to track TCP/IP flows within the network as well.

That's why we have NetFlow as an application for collecting IP traffic information. Cisco compares NetFlow informational reports to receiving a phone bill with detailed call information to track calls, call frequency, and even calls that shouldn't have been made at all! A more current analogy would be the IRS and certain additional government "alphabet agencies" watching who has talked to whom, when, and for how long!

Cisco IOS NetFlow efficiently provides a key set of services for IP applications, including network traffic accounting for baselining, usage-based network billing for consumers of network services, network design and planning, general network security, and DoS and DDoS monitoring capabilities as well as general network monitoring. Figure 3.15 shows basic flow monitoring via Cisco NetFlow with the latest version, version 9, which is called Flexible NetFlow.

FIGURE 3.15 Basic Flexible NetFlow



In Figure 3.15, let's assume that a host has connected to a server located in the fictitious Sales VLAN using Telnet. NetFlow can monitor the application by counting packets, bytes sent and received, and so on, and then send this information to a NetFlow collector.

NetFlow Overview and Flows

Understand that NetFlow is completely transparent to the users in the network, including all end stations and applications, and you don't need to run it on all your routers. Actually, you shouldn't because there's definitely overhead when using NetFlow because it requires memory for storing information in cache on the device. NetFlow enables near real-time visualization and analysis of recorded and aggregated flow data. You can specify the router, the aggregation scheme, and the time interval for when you want to view and then retrieve the relevant data and sort it into bar charts, pie charts, and so on. The components used with NetFlow include a router enabled with NetFlow and a NetFlow collector.

Service providers use NetFlow to do the following:

- Efficiently measuring who is using network service and for which purpose
- Accounting and charging back according to the resource utilizing level
- Using the measured information for more effective network planning so that resource allocation and deployment are well aligned with customer requirements
- Using the information to better structure and customize the set of available applications and services to meet user needs and customer service requirements

Moreover, there are different types of analyzers available to gather NetFlow statistics and analyze the traffic on your network by showing the following:

- Major users of the network, meaning top talkers, top listeners, top protocols, and so on
- Websites that are routinely visited, plus what's been downloaded
- Who's generating the most traffic and using excessive bandwidth
- Descriptions of bandwidth needs for an application as well as your available bandwidth

NetFlow is built around TCP/IP communication for statistical record-keeping using the concept of a flow. A flow is a unidirectional stream of packets between a source and destination host or system. With an understanding of TCP/IP, you can figure out that NetFlow is using socket information, meaning source and destination IP addresses and source and destination port numbers. But there are a few more fields that NetFlow uses. Here is a list of commonly used NetFlow flows:

- Source IP address
- Destination IP address
- Source port number
- Destination port number
- Layer 3 protocol field
- Type of Service (ToS) marking
- Input logical interface

As mentioned, the first four listings are the sockets used between the source and destination host, which identify the application. The protocol field identifies the data the packet is carrying, and ToS in the IPv4 header describes how QoS rules are applied to the packets in the flow. If a packet has a key field that's different from another packet, it's considered to belong to another flow. You configure NetFlow on the router's interfaces, and that's exactly what I'll show you next—how to configure and then verify NetFlow.

Configuring NetFlow

These four factors must be completed to properly implement NetFlow on a router:

- Configure NetFlow data capture by configuring ingress (incoming) and egress (outgoing) packets.
- Configure NetFlow data export by specifying the IP address of the NetFlow collector and the UDP port the collector listens for.
- Configure the NetFlow data export version by specifying the version of NetFlow, with version 9 being the most current.
- Verify NetFlow by analyzing the exported data on a host running a NetFlow collection engine or by using `show` command on the NetFlow router.

Here's an example of configuring NetFlow on the SF router:

```
SF(config)#int fa0/0
SF(config-if)#ip flow ingress
SF(config-if)#ip flow egress
SF(config-if)#exit
SF(config)#ip flow-export destination 172.16.20.254 9996
SF(config)#ip flow-export version ?
  1
  5
  9
SF(config)#ip flow-export version 9
SF(config)#ip flow-export source loopback 0
```

First I configured the Fast Ethernet 0/0 interface of the Corp router as both my ingress and egress interface, which tells the router to capture NetFlow data for flows on the interface. After that, I configured the NetFlow collector's IP address, as well as the version. Notice that I could opt to configure only versions 1, 5, and 9. Version 9 includes all the fields I mentioned already, plus MPLS and IPv6 information and ports. The loopback interface defines the source IP address of packets sent to the collector.

To verify NetFlow, you need to verify that the correct interfaces in the correct direction have been configured, starting with the `show ip flow interface` command like this:

```
SF#sh ip flow interface
FastEthernet0/0
```

```
  ip flow ingress
  ip flow egress
```

Sure enough! The correct interface of Fast Ethernet 0/0 is configured with the ingress and egress command. Now I'll check that I've correctly configured the export parameters via the `show ip flow export` command:

```
SF#sh ip flow export
Flow export v9 is enabled for main cache
  Exporting flows to 172.16.20.254 (9996) 172.16.20.254 (9996)
  Exporting using source interface Loopback0
  Version 9 flow records
  43 flows exported in 15 udp datagrams
[output cut]
```

Notice that the destination port is 9996. This is the Cisco default port number on which the NetFlow collectors listen for NetFlow packets. I can use the `sh ip cache flow` command to verify my flows by examining the information stored on a router directly, which will show that I'm actually collecting packets:

```
SF#sh ip cache flow
IP packet size distribution (161 total packets):
[output cut]
IP Flow Switching Cache, 278544 bytes
  1 active, 4095 inactive, 1 added
  215 ager polls, 0 flow alloc failures
  Active flows timeout in 30 minutes
  Inactive flows timeout in 15 seconds
IP Sub Flow Cache, 21640 bytes
  1 active, 1023 inactive, 1 added, 1 added to flow
  0 alloc failures, 0 force free
  1 chunk, 1 chunk added
  last clearing of statistics never
Protocol         Total    Flows   Packets Bytes  Packets Active(Sec) Idle(Sec)
--------         Flows     /Sec     /Flow  /Pkt     /Sec     /Flow     /Flow
TCP-Telnet       14         0.0      19     58       0.1      6.5      11.7
TCP-WWW           8         0.0       9    108       0.1      2.5       1.7
SrcIf         SrcIPaddress    DstIf      DstIPaddress    Pr SrcP DstP  Pkts
Fa0/0         172.16.10.1     gig0/1     255.255.255.255 11 0044 0050  1161
```

Visit `ccna .gg/ch3/b` for a companion MicroNugget from CBT Nuggets.

Nice: You can see that packets are truly being received—1161 so far—and the bottom lines show that the router is collecting flow for Telnet and HTTP. You can also see the source interface, source IP, destination interface, and source and destination ports in hex (50 is 80 in hex). It's important to remember that the `show ip cache flow` command provides a summary of the NetFlow statistics, including which protocols are in use.

Summary

Now you can implement and configure three different redundancy protocols! You learned that HSRP and VRRP are very similar but that VRRP is the open standard and can be used on any vendor's equipment. HSRP is Cisco proprietary and has a couple of features and configurations not available in VRRP—specifically, the preempt and interface tracking features.

You were then introduced to GLBP, which is also a Cisco proprietary protocol that can provide true load balancing for all of your clients, even if they are in the same subnet.

SNMP is an Application layer protocol that provides a message format for agents on a variety of devices to communicate to network management stations (NMSs). I discussed the basic information you need to use syslog and SNMP, that is, configuration and verification.

Last, I discussed Cisco NetFlow. Cisco IOS NetFlow efficiently provides a key set of services for IP applications, including network traffic accounting for baselining, usage-based network billing for consumers of network services, network design and planning, general network security, and DoS and DDoS monitoring capabilities, as well as general network monitoring.

Exam Essentials

Remember the three FHRPs. HSRP, VVRP, and GLBP are all FHRPs, with HSRP and GLBP being Cisco proprietary protocols.

Remember how load balancing works with HSRP and GLBP. HSRP load balance per VLAN's trunk links and GLBP can perform per-host load balancing.

Remember how to verify HSRP and GLBP. Use the `show standby` command with HSRP and `show glbp` with GLBP.

Remember the eight severity levels you can configure with the logging trap command. They are not listed in severity, but in alphabetical order, which makes it slightly more difficult to remember: 1=alerts, 2=critical, 7=debugging, 0=emergencies, 3=errors, 6=information, 5=notifications, 4=warnings.

Remember the differences between SNMPv2 and SNMPv3. SNMPv2 uses UDP but can use TCP, however, v2 still sends data to the NMS station in clear text. SNMPv3 uses TCP and authenticates users, plus can use ACLs in the SNMP strings to protect the NMS station from unauthorized use.

Understand what Cisco's NetFlow is used for. Cisco IOS NetFlow efficiently provides a key set of services for IP applications, including network traffic accounting for baselining, usage-based network billing for consumers of network services, network design and planning, general network security, and DoS and DDoS monitoring capabilities as well as general network monitoring.

Written Lab 3

The answers to this lab can be found in Appendix A, "Answers to Written Labs."

1. What command is used to verify your active router with HSRP?
2. Which command is used to verify your forwarding routers with GLBP?
3. Which FHRP can perform true load-balancing within the same subnet?
4. Which Cisco protocol can efficiently provide a key set of services for IP applications, including network traffic accounting?
5. Which syslog severity level results in notification-level messages?
6. Which protocol can request and receive information from a monitored device on the network?
7. Which syslog severity level results in warning-level messages?
8. Which command provides a summary of the NetFlow statistics, including which protocols are in use?
9. What command is used to configure a host to send messages to a syslog server?
10. Which router assigns a virtual MAC address to each member of the GLBP group?

Review Questions

NOTE The following questions are designed to test your understanding of this chapter's material. For more information on how to get additional questions, please see this book's introduction.

The answers to these questions can be found in Appendix B, "Answers to Chapter Review Questions."

1. How can you efficiently restrict the read-only function of a requesting SNMP management station based on the IP address?
   - **A.** Place an ACL on the logical control plane.
   - **B.** Place an ACL on the line when configuring the RO community string.
   - **C.** Place an ACL on the VTY line.
   - **D.** Place an ACL on all router interfaces.

2. Why would you use GLBP over HSRP and VRRP?
   - **A.** GLBP is an open standard protocol.
   - **B.** GLBP uses a virtual IP address, whereas HSRP and VRRP must have a static default gateway on each host.
   - **C.** GLBP provides true load-balancing within a single subnet.
   - **D.** GLBP is easily configured and propagated by DHCP to clients.
   - **E.** There is no reason to choose GLBP.

3. What is the default priority setting on an HSRP router?
   - **A.** 25
   - **B.** 50
   - **C.** 100
   - **D.** 125

4. You want to add a sequence number on your console message on a Cisco router. Which command will you use?
   - **A.** `service sequence-numbers`
   - **B.** `service timestamps`
   - **C.** `service number-sequence`
   - **D.** `sequence service messages`

5. You want to collect details about network traffic patterns on your network, including source and destination addresses and protocols used. Which of the following will you use?
   - **A.** `SNMPv3`
   - **B.** `Syslogv2`
   - **C.** `NetFlow 9`
   - **D.** `logging host` *`ip_address`*

6. You want to send a console message to a syslog server, but you only want to send status messages of 4 and lower. Which of the following commands will you use?
   - **A.** `logging trap emergencies`
   - **B.** `logging trap errors`
   - **C.** `logging trap debugging`
   - **D.** `logging trap notifications`
   - **E.** `logging trap critical`
   - **F.** `logging trap warnings`
   - **G.** `logging trap alerts`

7. In an FHRP network, which feature allows you to keep track of outside interfaces of an FHRP configured router?
   - **A.** Interface up/down status from `show ip int brief` command
   - **B.** `show nvram:show running-config` in RAM
   - **C.** Interface tracking within the HSRP configuration
   - **D.** HSRP's inability to verify or understand the status of outside interfaces

8. Which command enables you to view a summary of the NetFlow statistics of the protocols on a router?
   - **A.** `show ip flow`
   - **B.** `show ip cache flow`
   - **C.** `show ip netflow`
   - **D.** `show ip flow interface gi0/1`

9. Why should you use NetFlow over SNMP or syslog? (Choose three.)
   - **A.** NetFlow allows you to send configuration information to a remote network device.
   - **B.** NetFlow allows you to learn who is using which network resources.
   - **C.** NetFlow allows you to perform network capacity planning.
   - **D.** NetFlow allows you to retrieve accounting of network resource usage.
   - **E.** NetFlow receives alerts from remote routers if an interface fails.

10. You want to send a console message to a syslog server, but you only want to send status messages of 5 and lower. Which of the following commands will you use?

A. `logging trap emergencies`

B. `logging trap errors`

C. `logging trap debugging`

D. `logging trap notifications`

E. `logging trap critical`

F. `logging trap warnings`

G. `logging trap alerts`

11. Which two of the following are true regarding the output of the `show log` command? (Choose two.)

```
Corp#sh log
Syslog logging: enabled (11 messages dropped, 0 messages rate-limited,
      0 flushes, 0 overruns, xml disabled, filtering disabled)
  Console logging: level debugging, 3013 messages logged, xml disabled,
                   filtering disabled
  Monitor logging: level debugging, 0 messages logged, xml disabled,
                   filtering disabled
  Buffer logging: level debugging, 582 messages logged, xml disabled,
                   filtering disabled
  Logging Exception size (4096 bytes)
  Count and timestamp logging messages: disabled

  Trap logging: level debugging, 2523 message lines logged
  Logging to 172.16.10.1
```

A. The router is configured for trap level 6.

B. The router is configured for trap level 7.

C. Debugging messages will be sent only to 172.16.10.1.

D. Debugging messages will not be sent to the buffers.

12. What are the two pieces needed for NetFlow to gather data? (Choose two.)

A. An SNMP NMS station

B. Collector

C. Syslog configured

D. NetFlow-configured router

13. You want to send a console message to a syslog server, but you only want to send status messages of 3 and lower. Which of the following commands will you use?

A. `logging trap emergencies`

B. `logging trap errors`

C. `logging trap debugging`

D. `logging trap notifications`

E. `logging trap critical`

F. `logging trap warnings`

G. `logging trap alerts`

14. Which device will send the ARP replies to clients with GLBP?

A. The HSRP active router

B. The router with the highest IP address

C. The AVR router

D. The VRRP router

E. The passive listening router

15. Which command will you type in to see the active and listening router with GLBP?

A. `show standby`

B. `sh glbp`

C. `show active`

D. `show glbp infromation`

16. You want to send a console message to a syslog server, but you only want to send status messages of 7 and lower. Which of the following commands will you use?

A. `logging trap emergencies`

B. `logging trap errors`

C. `logging trap debugging`

D. `logging trap notifications`

E. `logging trap critical`

F. `logging trap warnings`

G. `logging trap alerts`

17. HSRP is load-balancing with three routers and three VLANs. Each router is configured with subinterfaces, one for each VLAN. How many routers in the group will forward traffic for each VLAN?

A. All the routers in the same group

B. Up to two routers per VLAN

C. One router per VLAN

D. HSRP must be configured with the GLBP command in order to do any load-balancing.

18. You want to send a console message to a syslog server, but you only want to send status messages of 6 and lower. Which of the following commands will you use?

A. `logging trap emergencies`

B. `logging trap errors`

C. `logging trap debugging`

D. `logging trap notifications`

E. `logging trap critical`

F. `logging trap warnings`

G. `logging trap informational`

19. Which SNMP version provides authentication, data integrity, and encryption?

A. SNMPv1

B. SNMPv2

C. SNMPv3

D. SNMPv4

E. SNMPv6

20. Which statements are true about both HSRP and GLBP? (Choose two.)

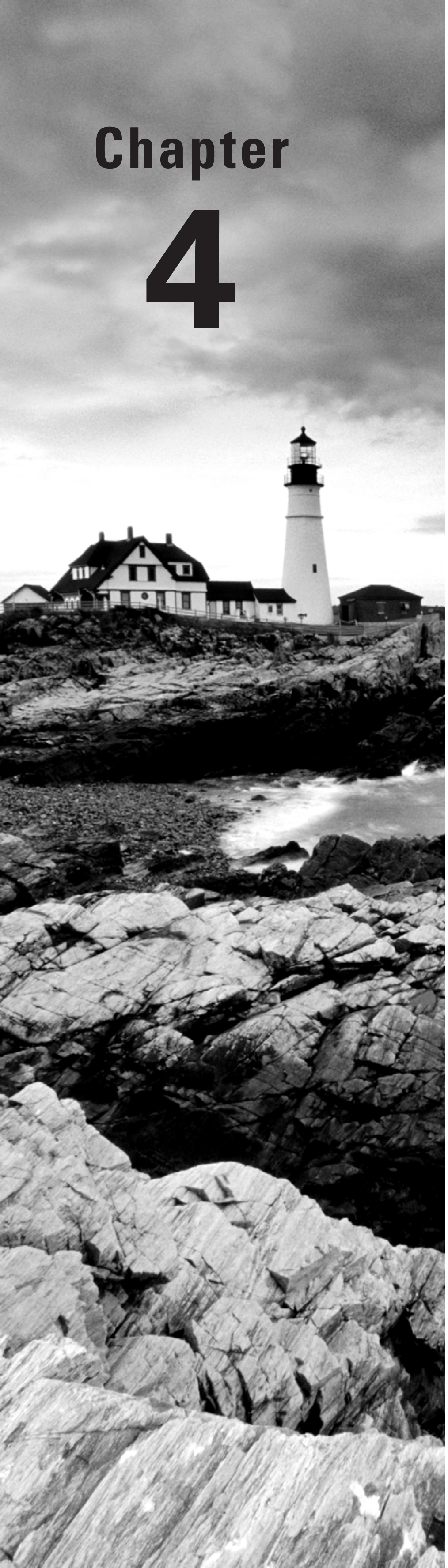
A. VRRP should always be used if possible instead of HSRP and GLBP.

B. HSRP is proprietary, and GLBP should be used with multiple vendors.

C. HSRP can load-balance based on VLANs for which they are active.

D. GLBP can have multiple forwarders with the same subnet/VLAN.

E. HSRP can load-balance on the same LAN.



Chapter

4

Troubleshooting IP, IPv6, and VLANs

THE FOLLOWING ICND2 EXAM TOPICS ARE COVERED IN THIS CHAPTER:

✓ **Troubleshooting**

- Identify and correct common network problems
- Troubleshoot and Resolve routing issues
    - Routing is enabled
    - Routing table is correct
    - Correct path selection
- Troubleshoot and resolve interVLAN routing problems
    - Connectivity
    - Encapsulation
    - Subnet
    - Native VLAN
    - Port mode trunk status

In this chapter, especially at first, it's going to seem like we're going over lot of the same ground and concepts already covered in other chapters. The reason for this is that troubleshooting is such a major focus of the Cisco ICND1 and ICND2 objectives that I've got to make sure I've guided you through this vital topic in depth. If not, then I just haven't done all I can to really set you up for success! So to make that happen, we're going to thoroughly examine troubleshooting with IP, IPv6, and *virtual LANs (VLANs)* now. And I can't stress the point enough that you absolutely must have a solid, fundamental understanding of IP and IPv6 routing as well as a complete understanding of VLANs and trunking nailed down tight if you're going to win at this!

To help you do that, I'll be using different scenarios to walk you through the Cisco troubleshooting steps to correctly solve the problems you're likely to be faced with. Although it's hard to tell exactly what the ICND1 and ICND2 exams will throw at you, you can read and completely understand the objectives so that no matter what, you'll be prepared, equipped, and up to the challenge. The way to do this is by building upon a really strong foundation, including being skilled at troubleshooting. This chapter is precisely designed, and exactly what you need, to seriously help solidify your troubleshooting foundation.

The chapters following this one will focus on EIGRP and OSPF, and each has its own troubleshooting section. Troubleshooting WAN protocols will be thoroughly covered in Chapter 7. In this chapter we'll concentrate solely on IP, IPv6, and VLAN troubleshooting.

NOTE To find up-to-the-minute updates for this chapter, please see `www.lammle.com/forum` or the book's web page at `www.sybex.com`.

Troubleshooting IP Network Connectivity

Let's start out by taking a moment for a short and sweet review of IP routing. Always remember that when a host wants to transmit a packet, IP looks at the destination address and determines if it's a local or remote request. If it's determined to be a local request, IP just broadcasts a frame out on the local network looking for the local host using an ARP request. If it's a remote request, the host sends an ARP request to the default gateway to discover the MAC address of the router.

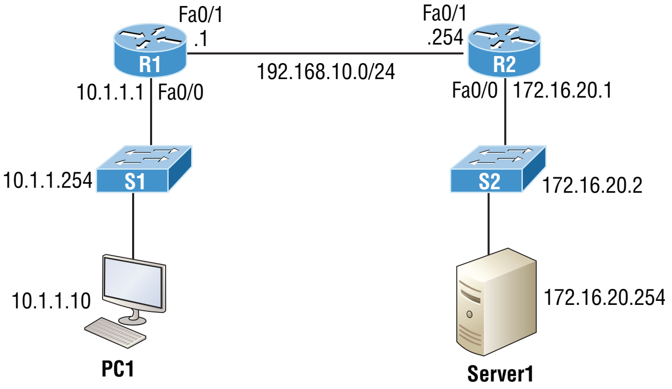
Once the hosts have the default gateway address, they'll send each packet that needs to be transmitted to the Data Link layer for framing, and newly framed packets are then sent out on the local collision domain. The router will receive the frame and remove the packet from the frame, and IP will then parse the routing table looking for the exit interface on the router. If the destination is found in the routing table, it will packet-switch the packet to the exit interface. At this point, the packet will be framed with new source and destination MAC addresses.

Okay, with that short review in mind, what would you say to someone who called you saying they weren't able to get to a server on a remote network? What's the first thing you would have this user do (besides reboot Windows) or that you would do yourself to test network connectivity? If you came up with using the Ping program, that's a great place to start. The Ping program is a great tool for finding out if a host is alive on the network with a simple ICMP echo request and echo reply. But being able to ping the host as well as the server doesn't guarantee that all is well in the network! Keep in mind that there's more to the Ping program than just being used as a quick and simple testing protocol.

To be prepared for the exam objectives, it's a great idea to get used to connecting to various routers and pinging from them. Of course, pinging from a router is not as good as pinging from the host reporting the problem, but that doesn't mean we can't isolate problems from the routers themselves.

Let's use Figure 4.1 as a basis to run through some troubleshooting scenarios.

FIGURE 4.1 Troubleshooting scenario



In this first scenario, a manager calls you and says that he cannot log in to Server1 from PC1. Your job is to find out why and fix it. The Cisco objectives are clear on the troubleshooting steps you need to take when a problem has been reported, and here they are:

1. Check the cables to find out if there's a faulty cable or interface in the mix and verify the interface's statistics.
2. Make sure that devices are determining the correct path from the source to the destination. Manipulate the routing information if needed.
3. Verify that the default gateway is correct.

4. Verify that name resolution settings are correct.
5. Verify that there are no *access control lists (ACLs)* blocking traffic.

In order to effectively troubleshoot this problem, we'll narrow down the possibilities by process of elimination. We'll start with PC1 and verify that it's configured correctly and also that IP is working correctly.

There are four steps for checking the PC1 configuration:

1. Test that the local IP stack is working by pinging the loopback address.
2. Test that the local IP stack is talking to the Data Link layer (LAN driver) by pinging the local IP address.
3. Test that the host is working on the LAN by pinging the default gateway.
4. Test that the host can get to remote networks by pinging remote Server1.

Let's check out the PC1 configuration by using the `ipconfig` command, or `ifconfig` on a Mac:

```
C:\Users\Todd Lammle>ipconfig


Windows IP Configuration


Ethernet adapter Local Area Connection:


   Connection-specific DNS Suffix  . : localdomain
   Link-local IPv6 Address . . . . . : fe80::64e3:76a2:541f:ebcb%11
   IPv4 Address. . . . . . . . . . . : 10.1.1.10
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 10.1.1.1
```

We can also check the route table on the host with the `route print` command to see if it truly does know the default gateway:

```
C:\Users\Todd Lammle>route print
[output cut]
IPv4 Route Table
=======================================================================
Active Routes:
Network Destination      Netmask        Gateway       Interface  Metric
          0.0.0.0        0.0.0.0        10.1.1.10     10.1.1.1   10
[output cut]
```

Between the output of the `ipconfig` command and the `route print` command, we can be assured that the hosts are aware of the correct default gateway.

NOTE For the Cisco objectives, it's extremely important to be able to check and verify the default gateway on a host and also that this address matches the router's interface!

So, let's verify that the local IP stack is initialized by pinging the loopback address now:

```
C:\Users\Todd Lammle>ping 127.0.0.1


Pinging 127.0.0.1 with 32 bytes of data:
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128


Ping statistics for 127.0.0.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

This first output confirms the IP address and configured default gateway of the host, and then I verified the fact that the local IP stack is working. Our next move is to verify that the IP stack is talking to the LAN driver by pinging the loopback address.

```
C:\Users\Todd Lammle>ping 10.1.1.10


Pinging 10.1.1.10 with 32 bytes of data:
Reply from 10.1.1.10: bytes=32 time<1ms TTL=128
Reply from 10.1.1.10: bytes=32 time<1ms TTL=128
Reply from 10.1.1.10: bytes=32 time<1ms TTL=128
Reply from 10.1.1.10: bytes=32 time<1ms TTL=128


Ping statistics for 10.1.1.10:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

And now that we know the local stack is solid and the IP stack is communicating to the LAN driver, it's time to check our local LAN connectivity by pinging the default gateway:

```
C:\Users\Todd Lammle>ping 10.1.1.1
```

```
Pinging 10.1.1.1 with 32 bytes of data:
Reply from 10.1.1.1: bytes=32 time<1ms TTL=128
Reply from 10.1.1.1: bytes=32 time<1ms TTL=128
Reply from 10.1.1.1: bytes=32 time<1ms TTL=128
Reply from 10.1.1.1: bytes=32 time<1ms TTL=128


Ping statistics for 10.1.1.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

Looking good! I'd say our host is in good shape. Let's try to ping the remote server next to see if our host is actually getting off the local LAN to communicate remotely:

```
C:\Users\Todd Lammle>ping 172.16.20.254


Pinging 172.16.20.254 with 32 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Request timed out.


Ping statistics for 172.16.20.254:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```

Well, looks like we've confirmed local connectivity but not remote connectivity, so we're going to have to dig deeper to isolate our problem. But first, and just as important, it's key to make note of what we can rule out at this point:

1. The PC is configured with the correct IP address and the local IP stack is working.
2. The default gateway is configured correctly and the PC's default gateway configuration matches the router interface IP address.
3. The local switch is working because we can ping through the switch to the router.
4. We don't have a local LAN issue, meaning our Physical layer is good because we can ping the router. If we couldn't ping the router, we would need to verify our physical cables and interfaces.

Let's see if we can narrow the problem down further using the `traceroute` command:

```
C:\Users\Todd Lammle>tracert 172.16.20.254
```

```
Tracing route to 172.16.20.254 over a maximum of 30 hops


  1     1 ms     1 ms    <1 ms  10.1.1.1
  2     *        *        *     Request timed out.
  3     *        *        *     Request timed out.
```

Well, we didn't get beyond our default gateway, so let's go over to R2 and see if we can talk locally to the server:

```
R2#ping 172.16.20.254


Pinging 172.16.20.254 with 32 bytes of data:
Reply from 172.16.20.254: bytes=32 time<1ms TTL=128
Reply from 172.16.20.254: bytes=32 time<1ms TTL=128
Reply from 172.16.20.254: bytes=32 time<1ms TTL=128
Reply from 172.16.20.254: bytes=32 time<1ms TTL=128


Ping statistics for 172.16.20.254:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```

Okay, we just eliminated a local LAN problem by connecting to Server1 from the R2 router, so we're good there. Let's summarize what we know so far:

1. PC1 is configured correctly.
2. The switch located on the 10.1.1.0 LAN is working.
3. PC1's default gateway is configured correctly.
4. R2 can communicate to the Server1 so we don't have a remote LAN issue.

But something is still clearly wrong, so what should we check now? Now would be a great time to verify the Server1 IP configuration and make sure the default gateway is configured correctly. Let's take a look:

```
C:\Users\Server1>ipconfig


Windows IP Configuration


Ethernet adapter Local Area Connection:


   Connection-specific DNS Suffix  . : localdomain
   Link-local IPv6 Address . . . . . : fe80::7723:76a2:e73c:2acb%11
   IPv4 Address. . . . . . . . . . . : 172.16.20.254
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 172.16.20.1
```

Okay—the Server1 configuration looks good and the R2 router can ping the server, so it seems that the server's local LAN is solid, the local switch is working, and there are no cable or interface issues. But let's zoom in on interface Fa0/0 on R2 and talk about what to expect if there were errors on this interface:

```
R2#sh int fa0/0
FastEthernet0/0 is up, line protocol is up
[output cut]
  Full-duplex, 100Mb/s, 100BaseTX/FX
  ARP type: ARPA, ARP Timeout 04:00:00
  Last input 00:00:05, output 00:00:01, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue: 0/40 (size/max)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     1325 packets input, 157823 bytes
     Received 1157 broadcasts (0 IP multicasts)
     0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
     0 watchdog
     0 input packets with dribble condition detected
     2294 packets output, 244630 bytes, 0 underruns
     0 output errors, 0 collisions, 3 interface resets
     347 unknown protocol drops
     0 babbles, 0 late collision, 0 deferred
     4 lost carrier, 0 no carrier
     0 output buffer failures, 0 output buffers swapped out
```

You've got to be able to analyze interface statistics to find problems there if they exist, so let's pick out the important factors relevant to meeting that challenge effectively now.

Speed and duplex settings Good to know that the most common cause of interface errors is a mismatched duplex mode between two ends of an Ethernet link. This is why it's so important to make sure that the switch and its hosts (PCs, router interfaces, etc.) have the same speed setting. If not, they just won't connect. And if they have mismatched duplex settings, you'll receive a legion of errors, which cause nasty performance issues, intermittent connectivity—even total loss of communication!

Using autonegotiation for speed and duplex is a very common practice, and it's enabled by default. But if this fails for some reason, you'll have to set the configuration manually like this:

```
Switch(config)#int gi0/1
Switch(config-if)#speed ?
  10    Force 10 Mbps operation
  100   Force 100 Mbps operation
  1000  Force 1000 Mbps operation
  auto  Enable AUTO speed configuration
Switch(config-if)#speed 1000
Switch(config-if)#duplex  ?
  auto  Enable AUTO duplex configuration
  full  Force full duplex operation
  half  Force half-duplex operation
Switch(config-if)#duplex  full
```

If you have a duplex mismatch, a telling sign is that the late collision counter will increment.

Input queue drops If the input queue drops counter increments, this signifies that more traffic is being delivered to the router that it can process. If this is consistently high, try to determine exactly when these counters are increasing and how the events relate to CPU usage. You'll see the ignored and throttle counters increment as well.

Output queue drops This counter indicates that packets were dropped due to interface congestion, leading to packet drops and queuing delays. When this occurs, applications like VoIP will experience performance issues. If you observe this constantly incrementing, consider QoS.

Input errors Input errors often indicate high errors such as CRCs. This can point to cabling problems, hardware issues, or duplex mismatches.

Output errors This is the total number of frames that the port tried to transmit when an issue such as a collision occurred.

We're going to move on in our troubleshooting process of elimination by analyzing the routers' actual configurations. Here's R1's routing table:

```
R1>sh ip route
[output cut]
Gateway of last resort is 192.168.10.254 to network 0.0.0.0


S*    0.0.0.0/0 [1/0] via 192.168.10.254
      10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C        10.1.1.0/24 is directly connected, FastEthernet0/0
L        10.1.1.1/32 is directly connected, FastEthernet0/0
```

```
      192.168.10.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.10.0/24 is directly connected, FastEthernet0/1
L        192.168.10.1/32 is directly connected, FastEthernet0/1
```

This actually looks pretty good! Both of our directly connected networks are in the table and we can confirm that we have a default route going to the R2 router. So now let's verify the connectivity to R2 from R1:

```
R1>sh ip int brief
Interface              IP-Address    OK? Method Status                Protocol
FastEthernet0/0        10.1.1.1      YES manual up                      up
FastEthernet0/1        192.168.10.1  YES manual up                      up
Serial0/0/0            unassigned    YES unset  administratively down down
Serial0/1/0            unassigned    YES unset  administratively down down
R1>ping 192.168.10.254
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.10.254, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms
```

This looks great too! Our interfaces are correctly configured with the right IP address and the Physical and Data Link layers are up. By the way, I also tested layer 3 connectivity by pinging the R2 Fa0/1 interface.

Since everything looks good so far, our next step is to check into the status of R2's interfaces:

```
R2>sh ip int brief
Interface              IP-Address      OK? Method Status                Protocol
FastEthernet0/0        172.16.20.1     YES manual up                      up
FastEthernet0/1        192.168.10.254  YES manual up                      up
R2>ping 192.168.10.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.10.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms
```

Well, everything still checks out at this point. The IP addresses are correct and the Physical and Data Link layers are up. I also tested the layer 3 connectivity with a ping to R1, so we're all good so far. We'll examine the routing table next:

```
R2>sh ip route
[output cut]
Gateway of last resort is not set

      10.0.0.0/24 is subnetted, 1 subnets
```

```
S        10.1.1.0 is directly connected, FastEthernet0/0
      172.16.0.0/16 is variably subnetted, 2 subnets, 2 masks
C        172.16.20.0/24 is directly connected, FastEthernet0/0
L        172.16.20.1/32 is directly connected, FastEthernet0/0
      192.168.10.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.10.0/24 is directly connected, FastEthernet0/1
L        192.168.10.254/32 is directly connected, FastEthernet0/1
```

Okay—we can see that all our local interfaces are in the table, as well as a static route to the 10.1.1.0 network. But do you see the problem? Look closely at the static route. The route was entered with an exit interface of Fa0/0, and the path to the 10.1.1.0 network is out Fa0/1! Aha! We've found our problem! Let's fix R2:

```
R2#config t
R2(config)#no ip route 10.1.1.0 255.255.255.0 fa0/0
R2(config)#ip route 10.1.1.0 255.255.255.0 192.168.10.1
```

That should do it. Let's verify from PC1:

```
C:\Users\Todd Lammle>ping 172.16.20.254


Pinging 172.16.20.254 with 32 bytes of data:
Reply from 172.16.20.254: bytes=32 time<1ms TTL=128
Reply from 172.16.20.254: bytes=32 time<1ms TTL=128
Reply from 172.16.20.254: bytes=32 time<1ms TTL=128
Reply from 172.16.20.254: bytes=32 time<1ms TTL=128


Ping statistics for 172.16.20.254
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

Our snag appears to be solved, but just to make sure, we really need to verify with a higher-level protocol like Telnet:

```
C:\Users\Todd Lammle>telnet 172.16.20.254
Connecting To 172.16.20.254...Could not open connection to the host, on
port 23: Connect failed
```

Okay, that's not good! We can ping to the Server1, but we can't telnet to it. In the past, I've verified that telnetting to this server worked, but it's still possible that we have a failure on the server side. To find out, let's verify our network first, starting at R1:

```
R1>ping 172.16.20.254
Type escape sequence to abort.
```

```
Sending 5, 100-byte ICMP Echos to 172.16.20.254, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/4 ms
R1>telnet 172.16.20.254
Trying 172.16.20.254 ...
% Destination unreachable; gateway or host down
```

This is some pretty ominous output! Let's try from R2 and see what happens:

```
R2#telnet 172.16.20.254
Trying 172.16.20.254 ... Open


User Access Verification


Password:
```

Oh my—I can ping the server from a remote network, but I can't telnet to it, but the local router R2 can! These factors eliminate the server being a problem since I can telnet to the server when I'm on the local LAN.

And we know we don't' have a routing problem because we fixed that already. So what's next? Let's check to see if there's an ACL on R2:

```
R2>sh access-lists
Extended IP access list 110
    10 permit icmp any any (25 matches)
```

Seriously? What a loopy access list to have on a router! This ridiculous list permits ICMP, but that's it. It denies everything except ICMP due to the implicit `deny ip any any` at the end of every ACL. But before we uncork the champagne, we need to see if this foolish list has been applied to our interfaces on R2 to confirm that this is really our problem:

```
R2>sh ip int fa0/0
FastEthernet0/0 is up, line protocol is up
  Internet address is 172.16.20.1/24
  Broadcast address is 255.255.255.255
  Address determined by setup command
  MTU is 1500 bytes
  Helper address is not set
  Directed broadcast forwarding is disabled
  Outgoing access list is 110
  Inbound  access list is not set
```

There it is—that's our problem all right! In case you're wondering why R2 could telnet to Server1, it's because an ACL filters only packets trying to go through the router—not packets generated at the router. Let's get to work and fix this:

```
R2#config t
R2(config)#no access-list 110
```

I just verified that I can telnet from PC1 to Server1, but let's try telnetting from R1 again:

```
R1#telnet 172.16.20.254
Trying 172.16.20.254 ... Open


User Access Verification


Password:
```

Nice—looks like we're set, but what about using the name?

```
R1#telnet Server1
Translating "Server1"...domain server (255.255.255.255)


% Bad IP address or host name
```

Well, we're not all set just yet. Let's fix R1 so that it can provide name resolution:

```
R1(config)#ip host Server1 172.16.20.254
R1(config)#^Z
R1#telnet Server1
Trying Server1 (172.16.20.254)... Open


User Access Verification


Password:
```

Great—things are looking good from the router, but if the customer can't telnet to the remote host using the name, we've got to check the DNS server to confirm connectivity and for the correct entry to the server. Another option would be to configure the local host table manually on PC1.

The last thing to do is to check the server to see if it's responding to HTTP requests via the `telnet` command, believe it or not! Here's an example:

```
R1#telnet 172.16.20.254 80
Trying 172.16.20.254, 80 ... Open
```

Yes—finally! Server1 is responding to requests on port 80, so we're in the clear. Now, let's mix things up a little by adding IPv6 to our network and work through the same troubleshooting steps.

Troubleshooting IPv6 Network Connectivity

Visit ccna.gg/ch4/a for a companion MicroNugget from CBT Nuggets.

I'm going to be straight with you: there isn't a lot that's going to be much different between this section and the process you just went through with the IPv4 troubleshooting steps. Except regarding the addressing of course! So other than that key factor, we'll take the same approach, using Figure 4.2, specifically because I really want to highlight the differences associated with IPv6. So the problem scenario I'm going to use will also stay the same: PC1 cannot communicate to Server1.

I want to point out that this is not an "introduction to IPv6" chapter, so I'm assuming you've got some IPv6 fundamentals down.

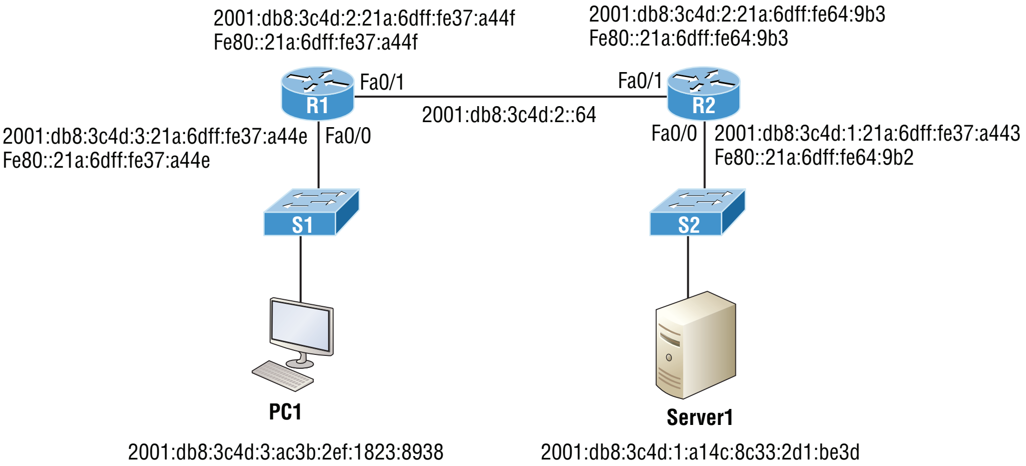


FIGURE 4.2 IPv6 troubleshooting scenario

2001:db8:3c4d:2:21a:6dff:fe37:a44f
Fe80::21a:6dff:fe37:a44f
2001:db8:3c4d:2:21a:6dff:fe64:9b3
Fe80::21a:6dff:fe64:9b3
Fa0/1
R1
2001:db8:3c4d:2::64
Fa0/1
R2
2001:db8:3c4d:3:21a:6dff:fe37:a44e
Fe80::21a:6dff:fe37:a44e
Fa0/0
Fa0/0
2001:db8:3c4d:1:21a:6dff:fe37:a443
Fe80::21a:6dff:fe64:9b2
S1
S2
PC1
Server1
2001:db8:3c4d:3:ac3b:2ef:1823:8938
2001:db8:3c4d:1:a14c:8c33:2d1:be3d

Notice that I documented both the *link-local* and *global addresses* assigned to each router interface in Figure 4.2. We need both in order to troubleshoot, so right away, you can see that things get a bit more complicated because of the longer addresses and the fact that there are multiple addresses per interface involved!

But *before* we start troubleshooting the IPv6 network in Figure 4.2, I want to refresh your memory on the ICMPv6 protocol, which is an important protocol in our troubleshooting arsenal.

ICMPv6

IPv4 used the ICMP workhorse for lots of tasks, including error messages like destination unreachable and troubleshooting functions like Ping and Traceroute. ICMPv6 still does those things for us, but unlike its predecessor, the v6 flavor isn't implemented as a separate layer 3 protocol. Instead, it's an integrated part of IPv6 and is carried after the basic IPv6 header information as an extension header.

ICMPv6 is used for router solicitation and advertisement, for neighbor solicitation and advertisement (i.e., finding the MAC addresses for IPv6 neighbors), and for redirecting the host to the best router (default gateway).

Neighbor Discovery (NDP)

ICMPv6 also takes over the task of finding the address of other devices on the local link. The Address Resolution Protocol is used to perform this function for IPv4, but that's been renamed Neighbor Discovery (ND or NDP) in ICMPv6. This process is now achieved via a multicast address called the solicited node address because all hosts join this multicast group upon connecting to the network.

Neighbor discovery enables these functions:

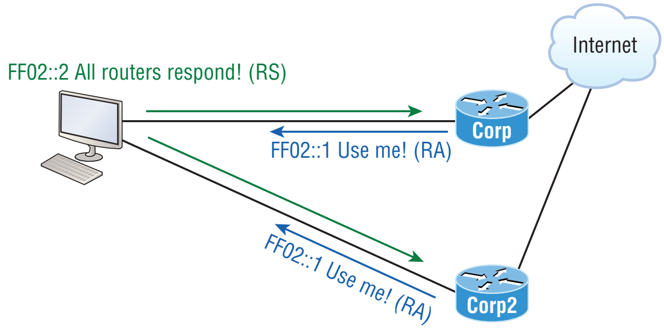
- Determining the MAC address of neighbors
- Router solicitation (RS) FF02::2
- Router advertisements (RA) FF02::1
- Neighbor solicitation (NS)
- Neighbor advertisement (NA)
- Duplicate address detection (DAD)

The part of the IPv6 address designated by the 24 bits farthest to the right is added to the end of the multicast address FF02:0:0:0:0:1:FF/104. When this address is queried, the corresponding host will send back its layer 2 address. Devices can find and keep track of other neighbor devices on the network in pretty much the same way. When I talked about RA and RS messages earlier in the CCENT chapters, and told you that they use multicast traffic to request and send address information, that too is actually a function of ICMPv6—specifically, neighbor discovery.

In IPv4, the protocol IGMP was used to allow a host device to tell its local router that it was joining a multicast group and would like to receive the traffic for that group. This IGMP function has been replaced by ICMPv6, and the process has been renamed multicast listener discovery.

With IPv4, our hosts could have only one default gateway configured, and if that router went down we had to fix the router, change the default gateway, or run some type of virtual default gateway with other protocols created as a solution for this inadequacy in IPv4. Figure 4.3 shows how IPv6 devices find their default gateways using neighbor discovery.

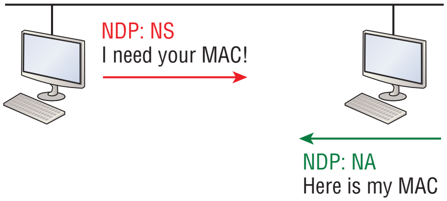
FIGURE 4.3 Router solicitation (RS) and router advertisement (RA)



IPv6 hosts send a router solicitation (RS) onto their data link asking for all routers to respond, and they use the multicast address FF02::2 to achieve this. Routers on the same link respond with a unicast to the requesting host, or with a router advertisement (RA) using FF02::1.

But that's not all! Hosts also can send solicitations and advertisements between themselves using a neighbor solicitation (NS) and neighbor advertisement (NA), as shown in Figure 4.4.

FIGURE 4.4 Neighbor solicitation (NS) and neighbor advertisement (NA)



Remember that RA and RS gather or provide information about routers and NS and NA gather information about hosts. Also, remember that a "neighbor" is a host on the same data link or VLAN.

With that foundation review in mind, here are the troubleshooting steps we'll progress through in our investigation:

1. Check the cables because there might be a faulty cable or interface. Verify interfacs statistics.
2. Make sure that devices are determining the correct path from the source to the destination. Manipulate the routing information if needed.
3. Verify that the default gateway is correct.

4. Verify that name resolution settings are correct, and especially for IPv6, make sure the DNS server is reachable via IPv4 and IPv6.
5. Verify that there are no ACLs that are blocking traffic.

In order to troubleshoot this problem, we'll use the same process of elimination, beginning with PC1. We must verify that it's configured correctly and that IP is working properly. Let's start by pinging the loopback address to verify the IPv6 stack:

```
C:\Users\Todd Lammle>ping ::1


Pinging ::1 with 32 bytes of data:
Reply from ::1: time<1ms
Reply from ::1: time<1ms
Reply from ::1: time<1ms
Reply from ::1: time<1ms
```

Well, the IPv6 stack checks out, so let's ping the Fa0/0 of R1, which PC1 is directly connected to on the same LAN, starting with the link-local address:

```
C:\Users\Todd Lammle>ping fe80::21a:6dff:fe37:a44e


Pinging fe80:21a:6dff:fe37:a44e with 32 bytes of data:
Reply from fe80::21a:6dff:fe37:a44e: time<1ms
Reply from fe80::21a:6dff:fe37:a44e: time<1ms
Reply from fe80::21a:6dff:fe37:a44e: time<1ms
Reply from fe80::21a:6dff:fe37:a44e: time<1ms
```

Next, we'll ping the global address on Fa0/0:

```
C:\Users\Todd Lammle>ping 2001:db8:3c4d:3:21a:6dff:fe37:a44e


Pinging 2001:db8:3c4d:3:21a:6dff:fe37:a44e with 32 bytes of data:
Reply from 2001:db8:3c4d:3:21a:6dff:fe37:a44e: time<1ms
Reply from 2001:db8:3c4d:3:21a:6dff:fe37:a44e: time<1ms
Reply from 2001:db8:3c4d:3:21a:6dff:fe37:a44e: time<1ms
Reply from 2001:db8:3c4d:3:21a:6dff:fe37:a44e: time<1ms
```

Okay—looks like PC1 is configured and working on the local LAN to the R1 router, so we've confirmed the Physical, Data Link, and Network layers between the PC1 and the R1 router Fa0/0 interface.

Our next move is to check the local connection on Server1 to the R2 router to verify that LAN. First we'll ping the link-local address of the router from Server1:

```
C:\Users\Server1>ping fe80::21a:6dff:fe64:9b2
```

```
Pinging fe80::21a:6dff:fe64:9b2  with 32 bytes of data:
Reply from fe80::21a:6dff:fe64:9b2: time<1ms
Reply from fe80::21a:6dff:fe64:9b2: time<1ms
Reply from fe80::21a:6dff:fe64:9b2: time<1ms
Reply from fe80::21a:6dff:fe64:9b2: time<1ms
```

And next, we'll ping the global address of Fa0/0 on R2:

```
C:\Users\Server1>ping 2001:db8:3c4d:1:21a:6dff:fe37:a443


Pinging 2001:db8:3c4d:1:21a:6dff:fe37:a443 with 32 bytes of data:
Reply from 2001:db8:3c4d:1:21a:6dff:fe37:a443: time<1ms
Reply from 2001:db8:3c4d:1:21a:6dff:fe37:a443: time<1ms
Reply from 2001:db8:3c4d:1:21a:6dff:fe37:a443: time<1ms
Reply from 2001:db8:3c4d:1:21a:6dff:fe37:a443: time<1ms
```

Let's quickly summarize what we know at this point:

1. By using the `ipconfig /all` command on PC1 and Server1, I was able to document their global and link-local IPv6 addresses.
2. We know the IPv6 link-local addresses of each router interface.
3. We know the IPv6 global address of each router interface.
4. We can ping from PC1 to router R1's Fa0/0 interface.
5. We can ping from Server1 to router R2's Fa0/0 interface.
6. We can eliminate a local problem on both LANs.

From here, we'll go to PC1 and see if we can route to Server1:

```
C:\Users\Todd Lammle>tracert 2001:db8:3c4d:1:a14c:8c33:2d1:be3d


Tracing route to 2001:db8:3c4d:1:a14c:8c33:2d1:be3d over a maximum of 30 hops


  1     Destination host unreachable.
```

Okay, now that's not good. Looks like we might have a routing problem. And on this little network, we're doing static IPv6 routing, so getting to the bottom of things will definitely take a little effort! But before we start looking into our potential routing issue, let's check the link between R1 and R2. We'll ping R2 from R1 to test the directly connected link.

The first thing you need to do before attempting to ping between routers is verify your addresses—yes, verify them again! Let's check out both routers, then try pinging from R1 to R2:

```
R1#sh ipv6 int brief
FastEthernet0/0            [up/up]
```

```
    FE80::21A:6DFF:FE37:A44E
    2001:DB8:3C4D:3:21A:6DFF:FE37:A44E
FastEthernet0/1            [up/up]
    FE80::21A:6DFF:FE37:A44F
    2001:DB8:3C4D:2:21A:6DFF:FE37:A44F
```

```
R2#sh ipv6 int brief
FastEthernet0/0            [up/up]
    FE80::21A:6DFF:FE64:9B2
    2001:DB8:3C4D:1:21A:6DFF:FE37:A443
FastEthernet0/1            [up/up]
    FE80::21A:6DFF:FE64:9B3
    2001:DB8:3C4D:2:21A:6DFF:FE64:9B3
```

```
R1#ping 2001:DB8:3C4D:2:21A:6DFF:FE64:9B3
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to ping 2001:DB8:3C4D:2:21A:6DFF:FE64:9B3, timeout
is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 0/2/8 ms
```

In the preceding output, you can see that I now have the IPv6 addresses for both the R1 and R2 directly connected interfaces. The output also shows that I used the Ping program to verify layer 3 connectivity. Just as with IPv4, we need to resolve the logical (IPv6) address to a MAC address in order to communicate on the local LAN. But unlike IPv4, IPv6 doesn't use ARP—it uses ICMPv6 neighbor solicitations instead—so after the successful ping, we can now see the neighbor resolution table on R1:

```
R1#sh ipv6 neighbors
IPv6 Address                              Age Link-layer Addr State Interface
FE80::21A:6DFF:FE64:9B3                     0 001a.6c46.9b09  DELAY Fa0/1
2001:DB8:3C4D:2:21A:6DFF:FE64:9B3           0 001a.6c46.9b09  REACH Fa0/1
```

Let's take a minute to talk about the possible states that a resolved address shows us:

INCMP (incomplete) Address resolution is being performed on the entry. A neighbor solicitation message has been sent, but the neighbor message has not yet been received.

REACH (reachable) Positive confirmation has been received confirming that the path to the neighbor is functioning correctly. REACH is good!

STALE The state is STALE when the interface has not communicated within the neighbor reachable time frame. The next time the neighbor communicates, the state will change back to REACH.

DELAY Occurs after the STALE state, when no reachability confirmation has been received within what's known as the DELAY_FIRST_PROBE_TIME. This means that the path was functioning but it hasn't had communication within the neighbor reachable time frame.

PROBE When in PROBE state, the configured interface is resending a neighbor solicitation and waiting for a reachability confirmation from a neighbor.

We can verify our default gateway with IPv6 with the `ipconfig` command like this:

```
C:\Users\Todd Lammle>ipconfig
   Connection-specific DNS Suffix  . : localdomain
   IPv6 Address. . . . . . . . . . . : 2001:db8:3c4d:3:ac3b:2ef:1823:8938
   Temporary IPv6 Address. . . . . . : 2001:db8:3c4d:3:2f33:44dd:211:1c3d
   Link-local IPv6 Address . . . . . : fe80::ac3b:2ef:1823:8938%11
   IPv4 Address. . . . . . . . . . . : 10.1.1.10
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : Fe80::21a:6dff:fe37:a44e%11
      10.1.1.1
```

It's important to understand that the default gateway will be the link-local address of the router, and in this case, we can see that the address the host learned is truly the link-local address of the Fa0/0 interface of R1. The `%11` is just used to identify an interface and isn't used as part of the IPv6 address.

Temporary IPv6 Addresses

The temporary IPv6 address, listed under the unicast IPv6 address as, "**2001:db8:3c4d:3:2f33:44dd:211:1c3d,**" was created by Windows to provide privacy from the EUI-64 format. This creates a global address from your host without using your MAC address by generating a random number for the interface and hashing it, which is then appended to the /64 prefix from the router. You can disable this feature with the following commands:

```
netsh interface ipv6 set global randomizeidentifiers=disabled
netsh interface ipv6 set privacy state-disabled
```

In addition to the `ipconfig` command, we can use the command `netsh interface ipv6 show neighbor` to verify our default gateway address:

```
C:\Users\Todd Lammle>netsh interface ipv6 show neighbor
[output cut]
```

```
Interface 11: Local Area Connection

Internet Address                              Physical Address   Type
--------------------------------------------  -----------------  -----------
2001:db8:3c4d:3:21a:6dff:fe37:a44e            00-1a-6d-37-a4-4e  (Router)
Fe80::21a:6dff:fe37:a44e                      00-1a-6d-37-a4-4e  (Router)
ff02::1                                       33-33-00-00-00-01  Permanent
ff02::2                                       33-33-00-00-00-02  Permanent
ff02::c                                       33-33-00-00-00-0c  Permanent
ff02::16                                      33-33-00-00-00-16  Permanent
ff02::fb                                      33-33-00-00-00-fb  Permanent
ff02::1:2                                     33-33-00-01-00-02  Permanent
ff02::1:3                                     33-33-00-01-00-03  Permanent
ff02::1:ff1f:ebcb                             33-33-ff-1f-eb-cb  Permanent
```

NOTE

I've checked the default gateway addresses on Server1 and they are correct. They should be, because this is provided directly from the router with an ICMPv6 RA (router advertisement) message. The output for that verification is not shown.

Let's establish the information we have right now:

1. Our PC1 and Server1 configurations are working and have been verified.
2. The LANs are working and verified, so there is no Physical layer issue.
3. The default gateways are correct.
4. The link between the R1 and R2 routers is working and verified.

So all this tells us is that it's now time to check our routing tables! We'll start with the R1 router:

```
R1#sh ipv6 route
C   2001:DB8:3C4D:2::/64 [0/0]
     via FastEthernet0/1, directly connected
L   2001:DB8:3C4D:2:21A:6DFF:FE37:A44F/128 [0/0]
     via FastEthernet0/1, receive
C   2001:DB8:3C4D:3::/64 [0/0]
     via FastEthernet0/0, directly connected
L   2001:DB8:3C4D:3:21A:6DFF:FE37:A44E/128 [0/0]
     via FastEthernet0/0, receive
L   FF00::/8 [0/0]
     via Null0, receive
```

All we can see in the output is the two directly connected interfaces configured on the router, and that won't help us send IPv6 packets to the 2001:db8:3c4d:1::/64 subnet off of Fa0/0 on R2. So let's find out what R2 can tell us:

```
R2#sh ipv6 route
C   2001:DB8:3C4D:1::/64 [0/0]
     via FastEthernet0/0, directly connected
L   2001:DB8:3C4D:1:21A:6DFF:FE37:A443/128 [0/0]
     via FastEthernet0/0, receive
C   2001:DB8:3C4D:2::/64 [0/0]
     via FastEthernet0/1, directly connected
L   2001:DB8:3C4D:2:21A:6DFF:FE64:9B3/128 [0/0]
     via FastEthernet0/1, receive
S   2001:DB8:3C4D:3::/64 [1/0]
     via 2001:DB8:3C4D:2:21B:D4FF:FE0A:539
L   FF00::/8 [0/0]
     via Null0, receive
```

Now we're talking—that tells us a lot more than R1's table did! We have both of our directly connected configured LANs, Fa0/0 and Fa0/1, right there in the routing table, as well as a static route to 2001:DB8:3C4D:3::/64, which is the remote LAN Fa0/0 off of R1, which is good. Now let's fix the route problem on R1 by adding a route that gives us access to the Server1 network and then move on to VLANs and trunking:

```
R1(config)#ipv6 route ::/0 fastethernet 0/1 FE80::21A:6DFF:FE64:9B3
```

I want to point out that I didn't need to make the default route as difficult as I did. I entered both the exit interface and next-hop link-local address when just the exit interface or next-hop global addresses would be mandatory, but not the link-local.

Next, we'll verify that we can now ping from PC1 to Server1:

```
C:\Users\Todd Lammle>ping 2001:db8:3c4d:1:a14c:8c33:2d1:be3d


Pinging 2001:db8:3c4d:1:a14c:8c33:2d1:be3d with 32 bytes of data:
Reply from 2001:db8:3c4d:1:a14c:8c33:2d1:be3d: time<1ms
Reply from 2001:db8:3c4d:1:a14c:8c33:2d1:be3d: time<1ms
Reply from 2001:db8:3c4d:1:a14c:8c33:2d1:be3d: time<1ms
Reply from 2001:db8:3c4d:1:a14c:8c33:2d1:be3d: time<1ms
```

Sweet—we're looking golden with this particular scenario! But know that it is still possible to have name resolution issues. If that were the case, you would just need to check your DNS server or local host table.

Moving on in the same way we did in the IPv4 troubleshooting section, it's a good time to check into your ACLs, especially if you're still having a problem after troubleshooting all your local LANs and all other potential routing issues. To do that, just use the `show ipv6 access-lists` command to verify all configured ACLs on a router and the `show ipv6 interface` command to verify if an ACL is attached to an interface. Once you've confirmed that your ACLs all make sense, you're good to go!

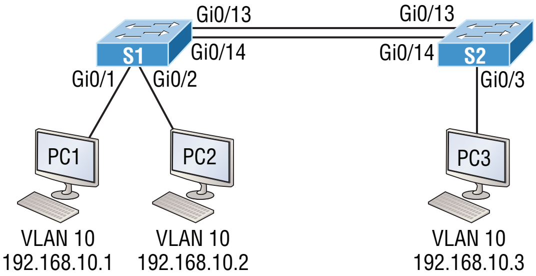
Troubleshooting VLAN Connectivity

You know by now that VLANs are used to break up broadcast domains in a layer 2 switched network. You've also learned that we assign ports on a switch into a VLAN broadcast domain by using the `switchport access vlan` command.

The access port carries traffic for a single VLAN that the port is a member of. If members of one VLAN want to communicate to members in the same VLAN that are located on a different switch, then a port between the two switches needs to be either configured to be a member of this single VLAN or configured as a trunk link, which passes information on all VLANs by default.

We're going to use Figure 4.5 to reference as we go through the procedures for troubleshooting VLAN and trunking.

FIGURE 4.5 VLAN connectivity



I'm going to begin with VLAN troubleshooting and then move on to trunk troubleshooting.

VLAN Troubleshooting

A couple of key times to troubleshoot VLANs are when and if you lose connectivity between hosts and when you're configuring new hosts into a VLAN but they're not working.

Here are the steps we'll follow to troubleshoot VLANs:

1. Verify the VLAN database on all your switches.
2. Verify your *content addressable memory (CAM)* table.
3. Verify that your port VLAN assignments are configured correctly.

And here's a list of the commands we'll be using in the following sections:

```
Show vlan
Show mac address-table
Show interfaces interface switchport
switchport access vlan vlan
```

VLAN Troubleshooting Scenario

A manager calls and says they can't communicate to the new sales team member that just connected to the network. How would you proceed to solve this issue? Well, because the sales hosts are in VLAN 10, we'll begin with step 1 and verify that our databases on both switches are correct.

First, I'll use the `show vlan` or `show vlan brief` command to check if the expected VLAN is actually in the database. Here's a look at the VLAN database on S1:

```
S1#sh vlan


VLAN Name                             Status    Ports
---- -------------------------------- --------- -------------------------------
1    default                          active    Gi0/3, Gi0/4, Gi0/5, Gi0/6
                                                Gi0/7, Gi0/8, Gi0/9, Gi0/10
                                                Gi0/11, Gi0/12, Gi0/13, Gi0/14
                                                Gi0/15, Gi0/16, Gi0/17, Gi0/18
                                                Gi0/19, Gi0/20, Gi0/21, Gi0/22
                                                Gi0/23, Gi0/24, Gi0/25, Gi0/26
                                                Gi0/27, Gi0/28
10   Sales                            active    Gi0/1, Gi0/2
20   Accounting                       active
26   Automation10                     active
27   VLAN0027                         active
30   Engineering                      active
170  VLAN0170                         active
501  Private501                       active
502  Private500                       active
[output cut]
```

This output shows that VLAN 10 is in the local database and that Gi0/1 and Gi0/2 are associated to VLAN 10.

So next, we'll go to step 2 and verify the CAM with the `show mac address-table` command:

```
S1#sh mac address-table
```

```
          Mac Address Table
-------------------------------------------


Vlan    Mac Address       Type        Ports
----    -----------       --------    -----
 All    0100.0ccc.cccc    STATIC      CPU
[output cut]
   1    000d.2830.2f00    DYNAMIC     Gi0/24
   1    0021.1c91.0d8d    DYNAMIC     Gi0/13
   1    0021.1c91.0d8e    DYNAMIC     Gi0/14
   1    b414.89d9.1882    DYNAMIC     Gi0/17
   1    b414.89d9.1883    DYNAMIC     Gi0/18
   1    ecc8.8202.8282    DYNAMIC     Gi0/15
   1    ecc8.8202.8283    DYNAMIC     Gi0/16
  10    001a.2f55.c9e8    DYNAMIC     Gi0/1
  10    001b.d40a.0538    DYNAMIC     Gi0/2
Total Mac Addresses for this criterion: 29
```

Okay—know that your switch will show quite a few MAC addresses assigned to the CPU at the top of the output; those MAC addresses are used by the switch to manage the ports. The very first MAC address listed is the base MAC address of the switch and used by STP in the bridge ID. In the preceding output, we can see that there there are two MAC addresses associated with VLAN 10 and that it was dynamically learned. We can also establish that this MAC address is associated to Gi0/1. S1 looks really good!

Let's take a look at S2 now. First, let's confirm that port PC3 is connected and check its configuration. I'll use the command `show interfaces` *`interface`* `switchport` command to do that:

```
S2#sh interfaces gi0/3 switchport
Name: Gi0/3
Switchport: Enabled
Administrative Mode: dynamic desirable
Operational Mode: static access
Administrative Trunking Encapsulation: negotiate
Operational Trunking Encapsulation: native
Negotiation of Trunking: On
Access Mode VLAN: 10 (Inactive)
Trunking Native Mode VLAN: 1 (default)
[output cut]
```

Okay—we can see that the port is enabled and that it's set to dynamic desirable. This means that if it connects to another Cisco switch, it will desire to trunk on that link. But keep in mind that we're using it as an access port, which is confirmed by the operational mode of static access. At the end of the output, the text shows `Access Mode VLAN: 10 (Inactive)`. This is not a good thing! Let's examine S2's CAM and see what we find out:

```
S2#sh mac address-table
          Mac Address Table
-------------------------------------------


Vlan    Mac Address       Type        Ports
----    -----------       --------    -----
 All    0100.0ccc.cccc    STATIC      CPU
 [output cut]
   1    001b.d40a.0538    DYNAMIC     Gi0/13
   1    0021.1bee.a70d    DYNAMIC     Gi0/13
   1    b414.89d9.1884    DYNAMIC     Gi0/17
   1    b414.89d9.1885    DYNAMIC     Gi0/18
   1    ecc8.8202.8285    DYNAMIC     Gi0/16
Total Mac Addresses for this criterion: 26
```

Referring back to Figure 4.5, we can see that the host is connected to Gi0/3. The problem here is that we don't see a MAC address dynamically associated to Gi0/3 in the MAC address table. So what do we know so far that can help us? Well first, we can see that Gi0/3 is configured into VLAN 10, but that VLAN is inactive. Second, the host off of Gi0/3 doesn't appear in the CAM table. Now would be a good time to take a look at the VLAN database like this:

```
S2#sh vlan brief


VLAN Name                             Status    Ports
---- -------------------------------- --------- -------------------------------
1    default                          active    Gi0/1, Gi0/2, Gi0/4, Gi0/5
                                                Gi0/6, Gi0/7, Gi0/8, Gi0/9
                                                Gi0/10, Gi0/11, Gi0/12, Gi0/13
                                                Gi0/14, Gi0/15, Gi0/16, Gi0/17
                                                Gi0/18, Gi0/19, Gi0/20, Gi0/21
                                                Gi0/22, Gi0/23, Gi0/24, Gi0/25
                                                Gi0/26, Gi0/27, Gi0/28
26   Automation10                     active
27   VLAN0027                         active
30   Engineering                      active
170  VLAN0170                         active
[output cut]
```

Look at that: there is no VLAN 10 in the database! Clearly the problem, but also an easy one to fix by simply creating the VLAN in the database:

```
S2#config t
S2(config)#vlan 10
S2(config-vlan)#name Sales
```

That's all there is to it. Now let's check the CAM again:

```
S2#sh mac address-table
          Mac Address Table
-------------------------------------------


Vlan    Mac Address       Type        Ports
----    -----------       --------    -----
 All    0100.0ccc.cccc    STATIC      CPU
[output cut]
   1    0021.1bee.a70d    DYNAMIC     Gi0/13
  10    001a.6c46.9b09    DYNAMIC     Gi0/3
Total Mac Addresses for this criterion: 22
```

We're good to go—the MAC address off of Gi0/3 shows in the MAC address table configured into VLAN 10.

That was pretty straightforward, but if the port had been assigned to the wrong VLAN, I would have used the `switch access vlan` command to correct the VLAN membership. Here's an example of how to do that:

```
S2#config t
S2(config)#int gi0/3
S2(config-if)#switchport access vlan 10
S2(config-if)#do sh vlan


VLAN Name                             Status    Ports
---- -------------------------------- --------- -------------------------------
1    default                          active    Gi0/1, Gi0/2, Gi0/4, Gi0/5
                                                Gi0/6, Gi0/7, Gi0/8, Gi0/9
                                                Gi0/10, Gi0/11, Gi0/12, Gi0/13
                                                Gi0/14, Gi0/15, Gi0/16, Gi0/17
                                                Gi0/18, Gi0/19, Gi0/20, Gi0/21
                                                Gi0/22, Gi0/23, Gi0/24, Gi0/25
                                                Gi0/26, Gi0/27, Gi0/28
10   Sales                            active    Gi0/3
```

Okay, great—we can see that our port Gi0/3 is in the VLAN 10 membership. Now let's try to ping from PC1 to PC3:

```
PC1#ping 192.168.10.3
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.10.3, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
```

No luck, so let's see if PC1 can ping PC2:

```
PC1#ping 192.168.10.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.10.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms
PC1#
```

That worked! I can ping a host that's a member of the same VLAN connected to the same switch, but I can't ping to a host on another switch that's a member of the same VLAN, which is VLAN 10. To get to the bottom of this, let's quickly summarize what we've learned so far:

1. We know that the VLAN database is now correct on each switch.
2. The MAC address table shows the ARP entries for each host as well as a connection to each switch.
3. We've verified that our VLAN memberships are now correct on all the ports we're using.

But since we still can't ping to a host on another switch, we need to start checking out the connections between our switches.

Trunk Troubleshooting

You'll need to troubleshoot trunk links when you lose connectivity between hosts that are in the same VLAN but are located on different switches. Cisco refers to this as "VLAN leaking." Seems to me we are leaking VLAN 10 between switches somehow.

These are the steps we'll take to troubleshoot VLANs:

1. Verify that the interface configuration is set to the correct trunk parameters.
2. Verify that the ports are configured correctly.
3. Verify the native VLAN on each switch.

And here are the commands we'll use to perform trunk troubleshooting:

```
Show interfaces trunk
Show vlan
Show interfaces interface trunk
Show interfaces interface switchport
Show dtp interface interface
switchport mode
switchport mode dynamic
switchport trunk native vlan vlan
```

Okay, let's get started by checking ports Gi0/13 and Gi0/14 on each switch because these are what the figure is showing as forming the connection between our switches. We'll start with the `show interfaces trunk` command:

```
S1>sh interfaces trunk


S2>sh interfaces trunk
```

Not a scrap of output—that's definitely a bad sign! Let's take another look at the `show vlan` output on S1 and see what we can find out:

```
S1>sh vlan brief


VLAN Name                             Status    Ports
---- -------------------------------- --------- -------------------------------
1    default                          active    Gi0/3, Gi0/4, Gi0/5, Gi0/6
                                                Gi0/7, Gi0/8, Gi0/9, Gi0/10
                                                Gi0/11, Gi0/12, Gi0/13, Gi0/14
                                                Gi0/15, Gi0/16, Gi0/17, Gi0/18
                                                Gi0/19, Gi0/20, Gi0/21, Gi0/22
                                                Gi0/23, Gi0/24, Gi0/25, Gi0/26
                                                Gi0/27, Gi0/28
10   Sales                            active    Gi0/1, Gi0/2
20   Accounting                       active
[output cut]
```

Nothing new from when we checked it a few minutes ago, but look there under VLAN 1—we can see interfaces Gi/013 and Gi0/14. This means that our ports between switches are members of VLAN 1 and will pass only VLAN 1 frames!

Typically I'll tell my students that if you type the `show vlan` command, you're really typing the nonexistent "show access ports" command since this output shows interfaces in access mode but doesn't show the trunk interfaces. This means that our ports between switches are access ports instead of trunk ports, so they'll pass information about only VLAN 1.

Let's go back over to the S2 switch to verify and see which port interfaces Gi0/13 and Gi0/14 are members of:

```
S2>sh vlan brief

VLAN Name                             Status    Ports
---- -------------------------------- --------- -------------------------------
1    default                          active    Gi0/1, Gi0/2, Gi0/4, Gi0/5
                                                Gi0/6, Gi0/7, Gi0/8, Gi0/9
                                                Gi0/10, Gi0/11, Gi0/12, Gi0/13
                                                Gi0/14, Gi0/15, Gi0/16, Gi0/17
                                                Gi0/18, Gi0/19, Gi0/20, Gi0/21
                                                Gi0/22, Gi0/23, Gi0/24, Gi0/25
                                                Gi0/26, Gi0/27, Gi0/28
10   Sales                            active    Gi0/3
```

Again, as with S1, the links between switches are showing in the output of the `show vlan` command, which means that they are not trunk ports. We can use the `show interfaces` *`interface`* `switchport` command to verify this as well:

```
S1#sho interfaces gi0/13 switchport
Name: Gi0/13
Switchport: Enabled
Administrative Mode: dynamic auto
Operational Mode: static access
Administrative Trunking Encapsulation: negotiate
Operational Trunking Encapsulation: native
Negotiation of Trunking: On
Access Mode VLAN: 1 (default)
Trunking Native Mode VLAN: 1 (default)
```

This output tells us that interface Gi0/13 is in dynamic auto mode. But its operational mode is static access, meaning it's not a trunk port. We can look closer at its trunking capabilities with the `show interfaces` *`interface`* `trunk` command:

```
S1#sh interfaces gi0/1 trunk

Port        Mode         Encapsulation  Status        Native vlan
Gi0/1       auto         negotiate      not-trunking  1
[output cut]
```

Sure enough—the port is not trunking, but we already knew that. Now we know it again. Notice that we can see that native VLAN is set to VLAN 1, which is the default native VLAN. This means that VLAN 1 is the default VLAN for untagged traffic.

Now, before we check the native VLAN on S2 to verify that there isn't a mismatch, I want to point out a key fact about trunking and how we would get these ports between switches to do that.

Many Cisco switches support the Cisco proprietary *Dynamic Trunking Protocol (DTP),* which is used to manage automatic trunk negotiation between switches. Cisco recommends that you don't allow this and to configure your switch ports manually instead. I agree with them!

Okay, with that in mind, let's check out our switch port Gi0/13 on S1 and view its DTP status. I'll use the `show dtp interface` *`interface`* command to view the DTP statistics:

```
S1#sh dtp interface gi0/13
DTP information for GigabitEthernet0/13:
  TOS/TAS/TNS:                              ACCESS/AUTO/ACCESS
  TOT/TAT/TNT:                              NATIVE/NEGOTIATE/NATIVE
  Neighbor address 1:                       00211C910D8D
  Neighbor address 2:                       000000000000
  Hello timer expiration (sec/state):       12/RUNNING
  Access timer expiration (sec/state):      never/STOPPED
```

Did you notice that our port GI0/13 from S1 to S2 is an access port configured to autonegotiate using DTP? That's interesting, and I want to delve a bit deeper into the different port configurations and how they affect trunking capabilities to clarify why.

Access Trunking is not allowed on a port set to access mode.

Auto Will trunk to neighbor switch only if the remote port is set to on or to desirable mode. This creates the trunk based on the DTP request from the neighboring switch.

Desirable This will trunk with all port modes except access. Ports set to dynamic desirable will communicate via DTP that the interface is attempting to become a trunk if the neighboring switch interface is able to become a trunk.

Nonegotiate No DTP frames are generated from the interface. Can only be used if the neighbor interface is manually set as trunk or access.

Trunk (on) Trunks with all switch port modes except access. Automatically enables trunking regardless of the state of the neighboring switch and regardless of any DTP requests.

Let's check out the different options available on the S1 switch with the `switchport mode dynamic` command:

```
S1(config-if)#switchport mode ?
  access        Set trunking mode to ACCESS unconditionally
  dot1q-tunnel  set trunking mode to TUNNEL unconditionally
  dynamic       Set trunking mode to dynamically negotiate access or trunk mode
  private-vlan  Set private-vlan mode
  trunk         Set trunking mode to TRUNK unconditionally
```

```
S1(config-if)#switchport mode dynamic ?
  auto       Set trunking mode dynamic negotiation parameter to AUTO
  desirable  Set trunking mode dynamic negotiation parameter to DESIRABLE
```

From interface mode, use the `switch mode trunk` command to turn trunking on. You can also use the `switch mode dynamic` command to set the port to auto or desirable trunking modes. To turn off DTP and any type of negotiation, use the `switchport nonegotiate` command.

Let's take a look at S2 and see if we can figure out why our two switches didn't create a trunk:

```
S2#sh int gi0/13 switchport
Name: Gi0/13
Switchport: Enabled
Administrative Mode: dynamic auto
Operational Mode: static access
Administrative Trunking Encapsulation: negotiate
Operational Trunking Encapsulation: native
Negotiation of Trunking: On
```

Okay—we can see that the port is in dynamic auto and that it's operating as an access port. Let's look into this further:

```
S2#sh dtp interface gi0/13
DTP information for GigabitEthernet0/3:
  DTP information for GigabitEthernet0/13:
  TOS/TAS/TNS:                              ACCESS/AUTO/ACCESS
  TOT/TAT/TNT:                              NATIVE/NEGOTIATE/NATIVE
  Neighbor address 1:                       000000000000
  Neighbor address 2:                       000000000000
  Hello timer expiration (sec/state):       17/RUNNING
  Access timer expiration (sec/state):      never/STOPPED
```

Do you see the problem? Don't be fooled—it's not that they're running in access mode; it's because two ports in dynamic auto will not form a trunk! This is a really common problem to look for since most Cisco switches ship in dynamic auto. The other issue you need to be aware of, as well as check for, is the frame-tagging method. Some switches run 802.1q, some run both 802.1q and *Inter-Switch Link (ISL) routing*, so be sure the tagging method is compatible between all of your switches!

It's time to fix our problem on the trunk ports between S1 and S2. All we need to do is to just fix one side of each link since dynamic auto will trunk with a port set to desirable or on:

```
S2(config)#int gi0/13
S2(config-if)#switchport mode dynamic desirable
23:11:37:%LINEPROTO-5-UPDOWN:Line protocol on Interface GigabitEthernet0/13,
changed state to down
```

```
23:11:37:%LINEPROTO-5-UPDOWN:Line protocol on Interface Vlan1, changed state to
down
23:11:40:%LINEPROTO-5-UPDOWN:Line protocol on Interface GigabitEthernet0/13,
changed state to up
23:12:10:%LINEPROTO-5-UPDOWN:Line protocol on Interface Vlan1, changed state to up
S2(config-if)#do show int trunk

Port        Mode         Encapsulation  Status        Native vlan
Gi0/13      desirable    n-isl          trunking      1
[output cut]
```

Nice—it worked! With one side in `Auto` and the other now in `Desirable`, DTPs will be exchanged and they will trunk. Notice in the preceding output that the mode of S2's Gi0/13 link is desirable and that the switches actually negotiated ISL as a trunk encapsulation—go figure! But don't forget to notice the native VLAN. We'll work on the frame-tagging method and native VLAN in a minute, but first, let's configure our other link:

```
S2(config-if)#int gi0/14
S2(config-if)#switchport mode dynamic desirable
23:12:%LINEPROTO-5-UPDOWN:Line protocol on Interface GigabitEthernet0/14, changed
state to down
23:12:%LINEPROTO-5-UPDOWN:Line protocol on Interface GigabitEthernet0/14, changed
state to up
S2(config-if)#do show int trunk

Port        Mode         Encapsulation  Status        Native vlan
Gi0/13      desirable    n-isl          trunking      1
Gi0/14      desirable    n-isl          trunking      1

Port        Vlans allowed on trunk
Gi0/13      1-4094
Gi0/14      1-4094
[output cut]
```

Great, we now have two trunked links between switches. But I've got to say, I really don't like the ISL method of frame tagging since it can't send untagged frames across the link. So let's change our native VLAN from the default of 1 to 392. The number 392 just randomly sounded good at the moment. Here's what I entered on S1:

```
S1(config-if)#switchport trunk native vlan 392
S1(config-if)#
23:17:40: Port is not 802.1Q trunk, no action
```

See what I mean? I tried to change the native VLAN and ISL basically responded with, "What's a native VLAN?" Very annoying, so I'm going to take care of that now!

```
S1(config-if)#int range gi0/13 - 14
S1(config-if-range)#switchport trunk encapsulation ?
  dot1q      Interface uses only 802.1q trunking encapsulation when trunking
  isl        Interface uses only ISL trunking encapsulation when trunking
  negotiate  Device will negotiate trunking encapsulation with peer on
             interface


S1(config-if-range)#switchport trunk encapsulation dot1q
23:23:%LINEPROTO-5-UPDOWN:Line protocol on Interface GigabitEthernet0/13, changed
state to down
23:23:%LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/14,
changed state to down
23:23:%CDP-4-NATIVE_VLAN_MISMATCH: Native VLAN mismatch discovered on
GigabitEthernet0/13 (392), with S2 GigabitEthernet0/13 (1).
23:23:%LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/14,
changed state to up
23:23:%LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/13,
changed state to up
23:23:%CDP-4-NATIVE_VLAN_MISMATCH: Native VLAN mismatch discovered on
GigabitEthernet0/13 (392), with S2 GigabitEthernet0/13 (1).
```

Okay, that's more like it! As soon as I changed the encapsulation type on S1, DTP frames changed the frame-tagging method between S2 to 802.1q. Since I had already changed the native VLAN on port Gi0/13 on S1, the switch lets us know, via CDP, that we now have a native VLAN mismatch. Let's proceed to deal with this by verifying our interfaces with the `show interface trunk` command:

```
S1#sh int trunk
Port        Mode         Encapsulation  Status        Native vlan
Gi0/13      auto         802.1q         trunking      392
Gi0/14      auto         802.1q         trunking      1


S2#sh int trunk

Port        Mode         Encapsulation  Status        Native vlan
Gi0/13      desirable    n-802.1q       trunking      1
Gi0/14      desirable    n-802.1q       trunking      1
```

Now notice that both links are running 802.1q and that S1 is in auto mode and S2 is in desirable mode. And we can see a native VLAN mismatch on port Gi0/13. We can also see the mismatched native VLAN with the `show interfaces` *`interface`* `switchport` command by looking at both sides of the link like this:

```
S2#sh interfaces gi0/13 switchport
Name: Gi0/13
Switchport: Enabled
Administrative Mode: dynamic desirable
Operational Mode: trunk
Administrative Trunking Encapsulation: negotiate
Operational Trunking Encapsulation: dot1q
Negotiation of Trunking: On
Access Mode VLAN: 1 (default)
Trunking Native Mode VLAN: 1 (default)


S1#sh int gi0/13 switchport
Name: Gi0/13
Switchport: Enabled
Administrative Mode: dynamic auto
Operational Mode: trunk
Administrative Trunking Encapsulation: dot1q
Operational Trunking Encapsulation: dot1q
Negotiation of Trunking: On
Access Mode VLAN: 1 (default)
Trunking Native Mode VLAN: 392 (Inactive)
```

So this has got to be bad, right? I mean really—are we sending any frames down that link or not? Let's see if we solved our little problem of not being able to ping to hosts from S1 to S2 and find out:

```
PC1#ping 192.168.10.3
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.10.3, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/4 ms
```

Yes, it works! Not so bad after all. We've solved our problem, or at least most of it. Having a native VLAN mismatch only means you can't send untagged frames down the link, which are essentially management frames like CDP, for example. So although it's not the end of the world, it will prevent us from being able to remotely manage the switch, or even sending any other types of traffic down just that one VLAN.

So am I saying you can just leave this issue the way it is? Well, you could, but you won't. No, you'll fix it because if you don't, CDP will send you a message every minute telling you that there's a mismatch, which will drive you mad! So, this is how we'll stop that from happening:

```
S2(config)#int gi0/13
S2(config-if)#switchport trunk native vlan 392
S2(config-if)#^Z
S2#sh int trunk

Port        Mode         Encapsulation  Status     Native vlan
Gi0/13      desirable    n-802.1q       trunking      392
Gi0/14      desirable    n-802.1q       trunking      1
[output cut]
```

All better! Both sides of the same link between switches are now using native VLAN 392 on Gigabit Ethernet 0/13. I want you to know that it's fine to have different native VLANs for each link if that's what works best for you. Each network is different and you have to make choices between options that will end up meeting your particular business requirements the most optimal way.

Visit ccna .gg/ch4/b for a companion MicroNugget from CBT Nuggets.

Summary

This chapter covered troubleshooting techniques from basic to advanced. Although most chapters in this book cover troubleshooting, this chapter focused purely on IPv4, IPv6, and VLAN/trunk troubleshooting.

You learned how to troubleshoot step-by-step from a host to a remote device. Starting with IPv4, you learned the steps to test the host and the local connectivity and then how to troubleshoot remote connectivity.

We then moved on to IPv6 and proceeded to troubleshoot using the same techniques that you learned with IPv4. It's important that you can use the verification commands that I used in each step of this chapter.

Last, I covered VLAN and trunk troubleshooting and how to go step-by-step through a switched network using verification commands and narrowing down the problem.

Exam Essentials

Remember the Cisco steps in troubleshooting an IPv4 and IPv6 network.

1. Check the cables to find out if there's a faulty cable or interface in the mix and verify the interface's statistics.
2. Make sure that devices are determining the correct path from the source to the destination. Manipulate the routing information if needed.
3. Verify that the default gateway is correct.
4. Verify that name resolution settings are correct.
5. Verify that there are no ACLs blocking traffic.

Remember the commands to verify and troubleshoot IPv4 and IPv6. You need to remember and practice the commands used in this chapter, especially `ping` and `traceroute` (`tracert` on Windows). But we also used the Windows commands `ipconfig` and `route print` and Cisco's commands `show ip int brief`, `show interface`, and `show route`.

Remember how to verify an ARP cache with IPv6. The command `show ipv6 neighbors` shows the IP-to-MAC-address resolution table on a Cisco router.

Remember to look at the statistics on a router and switch interface to determine problems. You've got to be able to analyze interface statistics to find problems if they exist, and this includes speed and duplex settings, input queue drops, output queue drops, and input and output errors.

Understand what a native VLAN is and how to change it. A native VLAN works with only 802.1q trunks and allows untagged traffic to traverse the trunk link. This is VLAN 1 by default on all Cisco switches, but it can be changed for security reasons with the `switchport native vlan` *`vlan`* command.

Written Lab 4

The answers to this lab can be found in Appendix A, "Answers to Written Labs."

Write the answers to the following questions:

1. If your IPv6 ARP cache shows an entry of INCMP, what does this mean?
2. You want traffic from VLAN 66 to traverse a trunked link untagged. Which command will you use?
3. What are the five modes you can set a switch port to?
4. You are having a network problem and have checked the cables to find out if there's a faulty cable or interface in the mix and also verified the interface's statistics, made sure that devices are determining the correct path from the source to the destination, and verified that you don't need to manipulate the routing. What are your next troubleshooting steps?
5. You need to find out if the local IPv6 stack is working on a host. What command will you use?

Hands-on Labs for Troubleshooting

Please check `www.lammle.com`, `www.lammlesim.com`, and `www.lammle.com/forum` for the latest information and downloads available for studying when using my books. Preconfigured hands-on troubleshooting labs are available for download, with the answers to the troubleshooting problems found on my forum. The troubleshooting hands-on lab simulator will be updated often.

Review Questions

NOTE The following questions are designed to test your understanding of this chapter's material. For more information on how to get additional questions, please see this book's introduction.

The answers to these questions can be found in Appendix B, "Answers to Chapter Review Questions."

1. You need to verify the IPv6 ARP cache on a router and see that the state of an entry is REACH. What does REACH mean?
   - A. The router is reaching out to get the address.
   - B. The entry is incomplete.
   - C. The entry has reached the end of life and will be discarded from the table.
   - D. A positive confirmation has been received by the neighbor and the path to it is functioning correctly.

2. What is the most common cause of interface errors?
   - A. Speed mismatch
   - B. Duplex mismatch
   - C. Buffer overflows
   - D. Collisions between a dedicated switch port and an NIC

3. Which command will verify the DTP status on a switch interface?
   - A. `sh dtp status`
   - B. `sh dtp status interface` *`interface`*
   - C. `sh interface` *`interface`* `dtp`
   - D. `sh dtp interface` *`interface`*

4. What mode will not allow DTP frames generated from a switch port?
   - A. Nonegotiate
   - B. Trunk
   - C. Access
   - D. Auto

5. The following output was generated by which command?

```
IPv6 Address                          Age Link-layer Addr State Interface
FE80::21A:6DFF:FE64:9B3                     0 001a.6c46.9b09  DELAY Fa0/1
2001:DB8:3C4D:2:21A:6DFF:FE64:9B3           0 001a.6c46.9b09  REACH Fa0/1
```

   A. `show ip arp`
   B. `show ipv6 arp`
   C. `show ip neighbors`
   D. `show ipv6 neighbors`

6. Which of the following states tells you that an interface has not communicated within the neighbor-reachable time frame?

   A. REACH
   B. STALE
   C. TIMEOUT
   D. CLEARED

7. You receive a call from a user who says that they cannot log in to a remote server, which only runs IPv6. Based on the output, what could the problem be?

```
C:\Users\Todd Lammle>ipconfig
   Connection-specific DNS Suffix  . : localdomain
   IPv6 Address. . . . . . . . . . . : 2001:db8:3c4d:3:ac3b:2ef:1823:8938
   Temporary IPv6 Address. . . . . . : 2001:db8:3c4d:3:2f33:44dd:211:1c3d
   Link-local IPv6 Address . . . . . : fe80::ac3b:2ef:1823:8938%11
   IPv4 Address. . . . . . . . . . . : 10.1.1.10
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 10.1.1.1
```

   A. The global address is in the wrong subnet.
   B. The IPv6 default gateway has not been configured or received from the router.
   C. The link-local address has not been resolved, so the host cannot communicate to the router.
   D. There are two IPv6 global addresses configured. One must be removed from the configuration.

8. Your host cannot reach remote networks. Based on the output, what is the problem?

```
C:\Users\Server1>ipconfig

Windows IP Configuration

Ethernet adapter Local Area Connection:

   Connection-specific DNS Suffix  . : localdomain
   Link-local IPv6 Address . . . . . : fe80::7723:76a2:e73c:2acb%11
   IPv4 Address. . . . . . . . . . . : 172.16.20.254
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 172.16.2.1
```

   A. The link-local IPv6 address is wrong.

   B. The IPv6 global address is missing.

   C. There is no DNS server configuration.

   D. The IPv4 default gateway address is misconfigured.

9. Which two commands will show you if you have a native VLAN mismatch?

   A. `show interface native vlan`

   B. `show interface trunk`

   C. `show interface` *`interface`* `switchport`

   D. `show switchport interface`

10. You connect two new Cisco 3560 switches together and expect them to use DTP and create a trunk. However, when you check statistics, you find that they are access ports and didn't negotiate. Why didn't DTP work on these Cisco switches?

   A. The ports on each side of the link are set to auto trunking.

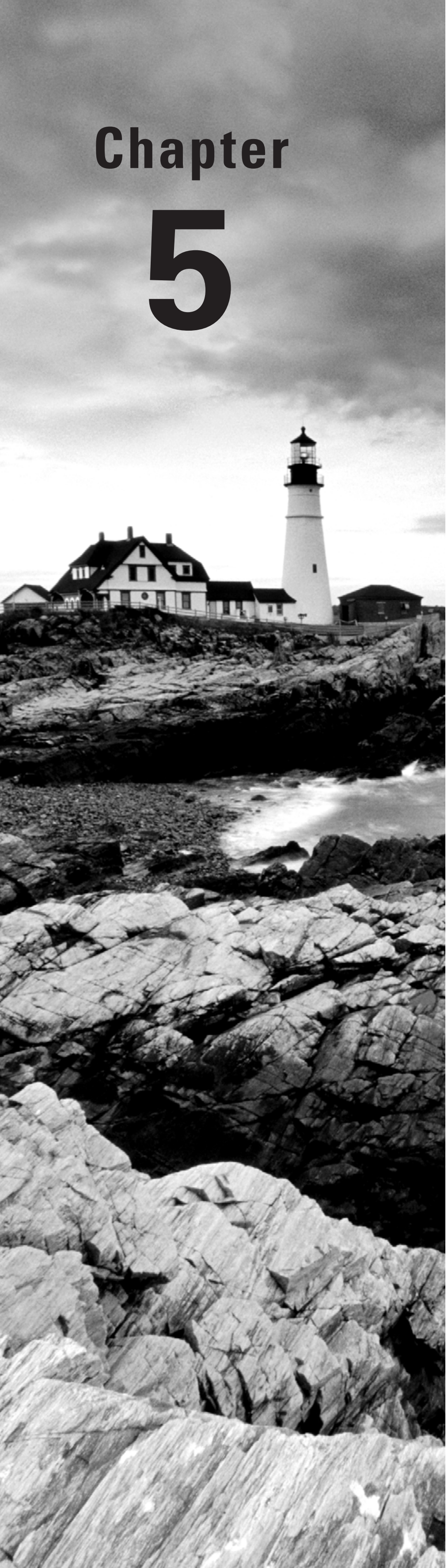
   B. The ports on each side of the link are set to on.

   C. The ports on each side of the link are set to dynamic.

   D. The ports on each side of the link are set to desirable.

Chapter

5



Enhanced IGRP

THE FOLLOWING ICND2 EXAM TOPICS ARE COVERED IN THIS CHAPTER:

✓ **IP Routing Technologies**

- Differentiate methods of routing and routing protocols
  - Administrative distance
  - Split horizon
  - Metric
  - Next hop
- Configure and verify EIGRP (single AS)
  - Feasible Distance/Feasible Successors/Administrative distance
  - Feasibility condition
  - Metric composition
  - Router ID
  - Auto-summary
  - Path selection
  - Load balancing
    - Equal
    - Unequal
  - Passive interface

✓ **Troubleshooting**

- Troubleshoot and resolve routing issues
  - Routing is enabled
  - Routing table is correct
  - Correct path selection
- Troubleshoot and resolve EIGRP problems
  - Neighbor adjacencies
  - AS number
  - Load balancing
  - Split horizon

Enhanced Interior Gateway Routing Protocol (EIGRP) is a Cisco protocol that runs on Cisco routers and on some Cisco switches. In this chapter, I'll cover the many features and functions of EIGRP, with an added focus on the unique way that it discovers, selects, and advertises routes.

EIGRP has a number of features that make it especially useful within large, complex networks. A real standout among these is its support of VLSM, which is crucial to its ultra-efficient scalability. EIGRP even includes benefits gained through other common protocols like OSPF and RIPv2, such as the ability to create route summaries at any location you choose.

I'll also cover key EIGRP configuration details and give you examples of each, as well as, demonstrate the various commands required to verify that EIGRP is working properly. Finally, I'll wrap up the chapter by showing you how to configure and verify EIGRPv6. I promise that after you get through it, you'll agree that EIGRPv6 is truly the easiest part of this chapter!

NOTE To find up-to-the-minute updates for this chapter, please see `www.lammle.com/forum` or the book's web page at `www.sybex.com`.

EIGRP Features and Operations

EIGRP is a classless, distance-vector protocol that uses the concept of an autonomous system to describe a set of contiguous routers that run the same routing protocol and share routing information, which also includes the subnet mask in its route updates. This is a very big deal because by advertising subnet information, this robust protocol enables us to use VLSM and permits summarization to be included within the design of EIGRP networks.

EIGRP is sometimes referred to as a *hybrid routing protocol* or an *advanced distance-vector protocol* because it has characteristics of both distance-vector and some link-state protocols. For example, EIGRP doesn't send link-state packets like OSPF does. Instead, it sends traditional distance-vector updates that include information about networks plus the cost of reaching them from the perspective of the advertising router.

EIGRP has link-state characteristics as well—it synchronizes network topology information between neighbors at startup and then sends specific updates only when topology changes occur (bounded updates). This particular feature is a huge advancement over RIP and is a big reason that EIGRP works so well in very large networks.

EIGRP has a default hop count of 100, with a maximum of 255, but don't let this confuse you because EIGRP doesn't rely on hop count as a metric like RIP does. In EIGRP-speak, hop count refers to how many routers an EIGRP route update packet can go through before it will be discarded, which limits the size of the autonomous system (AS). So don't forget that this isn't how metrics are calculated with EIGRP!

There are a bunch of powerful features that make EIGRP a real standout from other protocols. Here's a list of some of the major ones:

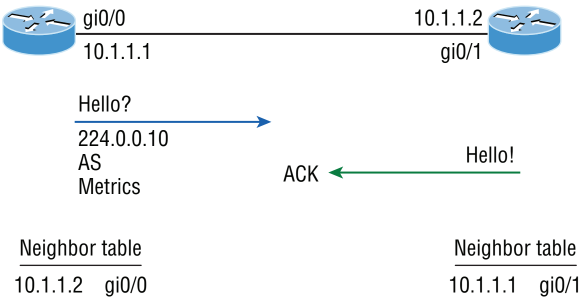
- Support for IP and IPv6 (and some other useless routed protocols) via protocol-dependent modules
- Considered classless (same as RIPv2 and OSPF)
- Support for VLSM/CIDR
- Support for summaries and discontiguous networks
- Efficient neighbor discovery
- Communication via Reliable Transport Protocol (RTP)
- Best path selection via Diffusing Update Algorithm (DUAL)
- Reduced bandwidth usage with bounded updates
- No broadcasts

NOTE Cisco refers to EIGRP as a distance-vector routing protocol but also as an advanced distance-vector or even a hybrid routing protocol.

Neighbor Discovery

Before EIGRP routers can exchange routes with each other, they must become neighbors, and there are three conditions that must be met before this can happen, as shown in Figure 5.1.

FIGURE 5.1 EIGRP neighbor discovery



And these three things will be exchanged with directly connected neighbors:

- Hello or ACK received
- AS numbers match
- Identical metrics (K values)

Link-state protocols often use Hello messages to establish who their neighbors are because they usually don't send out periodic route updates but still need a way to help neighbors know when a new peer has arrived or an old one has gone down. And because Hellos are also used to maintain neighbor relationships, it follows that EIGRP routers must also continuously receive Hellos from their neighbors.

But EIGRP routers that belong to different ASs don't automatically share routing information and, therefore, don't become neighbors. This factor is really helpful operating in larger networks because it reduces the amount of route information propagated through a specific AS. But it also means that manual redistribution can sometimes be required between different ASs as a result. Because metrics play a big role in choosing between the five possible factors to be evaluated when choosing the best possible route, it's important that all EIGRP neighbors agree on how a specific route is chosen. This is vital because the calculations on one router depend upon the calculations of its neighbors.

Hellos between EIGRP routers are set to 5 seconds by default. Another timer that's related to the *hello timer* is the *hold timer*. The hold timer determines the amount of time a router is willing to wait to get a Hello from a neighbor before declaring it dead. Once a neighbor is declared dead, it's removed from the neighbor table and all routes that depended upon it are recalculated. Interestingly, the hold timer configuration doesn't determine how long a router waits before it declares neighbors dead; it establishes how long the router will tell others to wait before they can declare it dead. This means that the hold timers on neighboring routers don't need to match because they only tell the others how long to wait.

The only time EIGRP advertises its entire information is when it discovers a new neighbor and forms a relationship or adjacency with it by exchanging Hello packets. When this happens, both neighbors then advertise their complete information to one another. After each has learned its neighbor's routes, only changes to the routing table will be propagated.

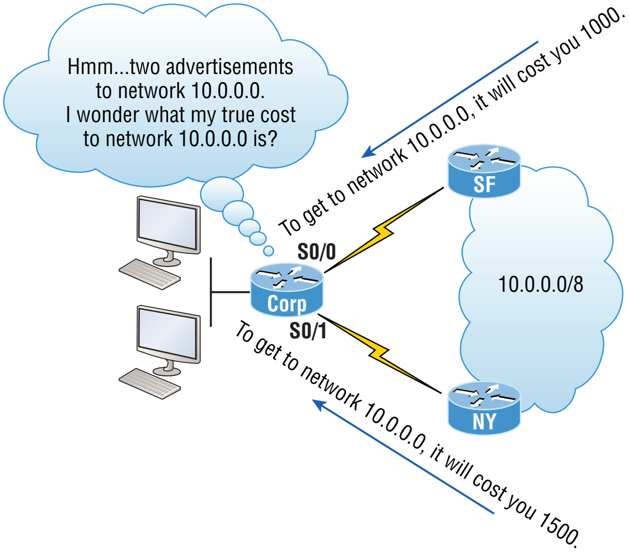
During each EIGRP session running on a router, a neighbor table is created in which the router stores information about all routers known to be directly connected neighbors. Each neighboring router's IP address, hold time interval, *smooth round-trip timer (SRTT)*, and queue information are all kept in this table. It's an important reference used to establish that topology changes have occurred that neighboring routers need to know about.

To sum this all up, remember that EIGRP routers receive their neighbors' updates and store them in a local topology table that contains all known routes from all known neighbors and serves as the raw material from which the best routes are selected.

Let's define some terms before we move on:

Reported/advertised distance (AD) This is the metric of a remote network, as reported by a neighbor. It's also the routing table metric of the neighbor and is the same as the second number in parentheses as displayed in the topology table. The first number is the administrative distance, and I'll discuss more about these values in a minute. In Figure 5.2, routers SF and NY are both advertising the path to network 10.0.0.0 to the Corp router, but the cost through SF to network 10.0.0.0 is less than NY.

FIGURE 5.2 Advertised distance


Hmm...two advertisements to network 10.0.0.0. I wonder what my true cost to network 10.0.0.0 is?
To get to network 10.0.0.0, it will cost you 1000.
SF
S0/0
Corp
S0/1
10.0.0.0/8
NY
To get to network 10.0.0.0, it will cost you 1500.

We're not done yet because the Corp router still needs to calculate its cost to each neighbor.

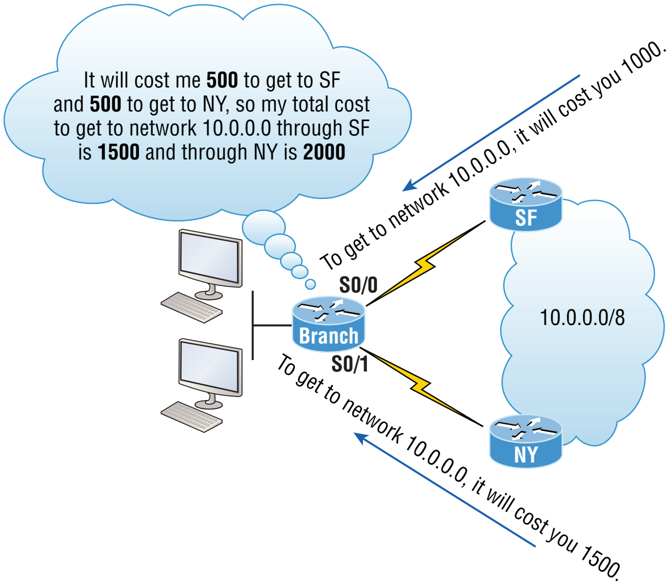
Feasible distance (FD) This is the best metric among all paths to a remote network, including the metric to the neighbor that's advertising the remote network. The route with the lowest FD is the route that you'll find in the routing table because it's considered the best path. The metric of a feasible distance is calculated using the metric reported by the neighbor that's referred to as the reported or advertised distance plus the metric to the neighbor reporting the route. In Figure 5.3, the Corp router will have the path through router SF to network 10.0.0.0 in the routing table since its the lowest feasible distance. It's the lowest true cost from end to end.

Take a look at an EIGRP route that's been injected into a routing table and find the FD listed in the entry.

```
D    10.0.0.0/8 [90/2195456] via 172.16.10.2, 00:27:06,Serial0/0
```

First, the `D` means Dual, and is an EIGRP injected route and is the route used by EIGRP to forward traffic to the 10.0.0.0 network via its neighbor, 172.16.10.2. But that's not what I want to focus on right now. See the `[90/2195456]` entry in the line? The first number (`90`) is the administrative distance (AD), which is not to be confused with advertised distance (AD), which is why a lot of people call it the reported distance! The second number, is the feasible distance (FD), or the entire cost for this router to get to network 10.0.0.0. To sum this up, the neighbor router sends a reported, or advertised, distance (RD/AD) for network 10.0.0.0, and EIGRP calculates the cost to get to that neighbor and then adds those two numbers together to get the FD, or total cost.

FIGURE 5.3 Feasible distance


It will cost me 500 to get to SF and 500 to get to NY, so my total cost to get to network 10.0.0.0 through SF is 1500 and through NY is 2000
To get to network 10.0.0.0, it will cost you 1000.
SF
S0/0
Branch
S0/1
10.0.0.0/8
NY
To get to network 10.0.0.0, it will cost you 1500.

Neighbor table Each router keeps state information about adjacent neighbors. When a newly discovered neighbor is found, its address and interface are recorded and the information is held in the neighbor table, stored in RAM. Sequence numbers are used to match acknowledgments with update packets. The last sequence number received from the neighbor is recorded so that out-of-order packets can be detected. We'll get into this more, later in the chapter, when we look at the neighbor table and find out how it's useful for troubleshooting links between neighbor routers.

Topology table The topology table is populated by the neighbor table and the Diffusing Update Algorithm (DUAL) calculates the best loop-free path to each remote network. It contains all destinations advertised by neighboring routers, holding each destination address and a list of neighbors that have advertised the destination. For each neighbor, the advertised metric (distance), which comes only from the neighbor's routing table, is recorded, as well as, the FD. The best path to each remote network is copied and placed in the routing table and then IP will use this route to forward traffic to the remote network. The path copied to the routing table is called a successor router—think "successful" to help you remember. The path with a good, but less desirable, cost will be entered in the topology table as a backup link and called the feasible successor. Let's talk more about these terms now.

NOTE

The neighbor and topology tables are stored in RAM and maintained through the use of Hello and update packets. While the routing table is also stored in RAM, the information stored in the routing table is gathered only from the topology table.

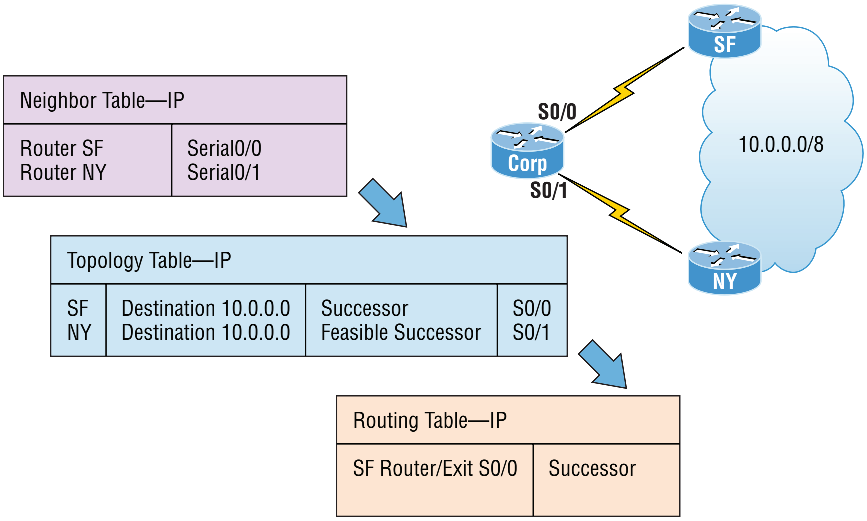
Feasible successor (FS) So a feasible successor is basically an entry in the topology table that represents a path that's inferior to the successor route(s). An FS is defined as a path whose advertised distance is less than the feasible distance of the current successor and considered a backup route. EIGRP will keep up to 32 feasible successors in the topology table in 15.0 code but only up to 16 in previous IOS versions, which is still a lot! Only the path with the best metric—the successor—is copied and placed in the routing table. The `show ip eigrp topology` command will display all the EIGRP feasible successor routes known to the router.

NOTE A feasible successor is a backup route and is stored in the topology table. A successor route is stored in the topology table and is copied and placed in the routing table.

Successor A successor route—again, think "successful"—is the best route to a remote network. A successor route is the lowest cost to a destination and stored in the topology table along with everything else. However, this particular best route is copied and placed in the routing table so IP can use it to get to the remote network. The successor route is backed up by a feasible successor route, which is also stored in the topology table, if there's one available. The routing table contains only successor routes; the topology table contains successor and feasible successor routes.

Figure 5.4 illustrates that the SF and NY routers each have subnets of the 10.0.0.0 network and the Corp router has two paths to get to this network.

FIGURE 5.4 The tables used by EIGRP



Neighbor Table—IP	
Router SF Router NY	Serial0/0 Serial0/1

Topology Table—IP			
SF	Destination 10.0.0.0	Successor	S0/0
NY	Destination 10.0.0.0	Feasible Successor	S0/1

Routing Table—IP	
SF Router/Exit S0/0	Successor

As shown in Figure 5.4, there are two paths to network 10.0.0.0 that can be used by the Corp router. EIGRP picks the best path and places it in the routing table, but if both links have equal-cost paths, EIGRP would load-balance between them—up to four links, by default. By using the successor, and having feasible successors in the topology table as backup links, the network can converge instantly and updates to any neighbor make up the only traffic sent from EIGRP—very clean!

Reliable Transport Protocol (RTP)

EIGRP depends on a proprietary protocol, called *Reliable Transport Protocol (RTP)*, to manage the communication of messages between EIGRP-speaking routers. As the name suggests, reliability is a key concern of this protocol, so Cisco designed this mechanism, which leverages multicasts and unicasts, to ensure that updates are delivered quickly and that data reception is tracked accurately.

But how does this really work? Well, when EIGRP sends multicast traffic, it uses the Class D address 224.0.0.10, and each EIGRP router knows who its neighbors are. For each multicast it sends out, a list is built and maintained that includes all the neighbors who have replied. If a router doesn't get a reply from a neighbor via the multicast, EIGRP will then try using unicasts to resend the same data. If there's no reply from a neighbor after 16 unicast attempts, that neighbor will then be declared dead. This process is often referred to as *reliable multicast*.

Routers keep track of the information they send by assigning a sequence number to each packet that enables them to identify old, redundant information and data that's out of sequence. You'll get to actually see this information in the neighbor table coming up when we get into configuring EIGRP.

Remember, EIGRP is all about topology changes and updates, making it the quiet, performance-optimizing protocol it is. Its ability to synchronize routing databases at startup time, while maintaining the consistency of databases over time, is achieved quietly by communicating only necessary changes. The downside here is that you can end up with a corrupted routing database if any packets have been permanently lost or if packets have been mishandled out of order!

Here's a description of the five different types of packets used by EIGRP:

Update An *Update packet* contains route information. When these are sent in response to metric or topology changes, they use reliable multicasts. In the event that only one router needs an update, like when a new neighbor is discovered, it's sent via unicasts. Keep in mind that the unicast method still requires an acknowledgment, so updates are always reliable regardless of their underlying delivery mechanism.

Query A *Query packet* is a request for specific routes and always uses the reliable multicast method. Routers send queries when they realize they've lost the path to a particular network and are searching for alternatives.

Reply A *Reply packet* is sent in response to a query via the unicast method. Replies either include a specific route to the queried destination or declare that there's no known route.

Hello A *Hello packet* is used to discover EIGRP neighbors and is sent via unreliable multicast, meaning it doesn't require an acknowledgment.

ACK An *ACK packet* is sent in response to an update and is always unicast. ACKs are never sent reliably because this would require another ACK sent for acknowledgment, which would just create a ton of useless traffic!

It's helpful to think of all these different packet types like envelopes. They're really just types of containers that EIGRP routers use to communicate with their neighbors. What's really interesting is the actual content envelopes these communications and the procedures that guide their conversations, and that's what we'll be exploring next!

Diffusing Update Algorithm (DUAL)

I mentioned that EIGRP uses *Diffusing Update Algorithm (DUAL)* for selecting and maintaining the best path to each remote network. DUAL allows EIGRP to carry out these vital tasks:

- Figure out a backup route if there's one available.
- Support variable length subnet masks (VLSMs).
- Perform dynamic route recoveries.
- Query neighbors for unknown alternate routes.
- Send out queries for an alternate route.

Quite an impressive list, but what really makes DUAL so great is that it enables EIGRP to converge amazingly fast! The key to the speed is twofold: First, EIGRP routers maintain a copy of all of their neighbors' routes to refer to for calculating their own cost to each remote network. So if the best path goes down, all it often takes to find another one is a quick scan of the topology table looking for a feasible successor. Second, if that quick table survey doesn't work out, EIGRP routers immediately ask their neighbors for help finding the best path. It's exactly this, ahem, DUAL strategy of reliance upon, and the leveraging of, other routers' information that accounts for the algorithm's "diffusing" character. Unlike other routing protocols where the change is propagated through the entire network, EIGRP bounded updates are propagated only as far as needed.

Three critical conditions must be met for DUAL to work properly:

- Neighbors are discovered or noted as dead within a finite time.
- All transmitted messages are received correctly.
- All changes and messages are processed in the order in which they're detected.

As you already know, the Hello protocol ensures the rapid detection of new or dead neighbors, and RTP provides a reliable method of conveying and sequencing messages. Based upon this solid foundation, DUAL can then select and maintain information about the best paths. Let's check further into the process of route discovery and maintenance next.

Route Discovery and Maintenance

The hybrid nature of EIGRP is fully revealed in its approach to route discovery and maintenance. Like many link-state protocols, EIGRP supports the concept of neighbors that are formally discovered via a Hello process and whose state is monitored thereafter. And like many distance-vector protocols, EIGRP uses the routing-by-rumor approach, which implies that many routers within an AS never actually hear about a route update firsthand. Instead, these devices rely on “network gossip” to hear about neighbors and their respective status via another router that may have also gotten the info from yet another router and so on.

Given all of the information that EIGRP routers have to collect, it follows that they must have a place to store it, and they do this in the tables I referred to earlier in this chapter. As you know, EIGRP doesn’t depend on just one table—it actually uses three of them to store important information about its environment:

Neighbor table Contains information about the specific routers with whom neighbor relationships have been formed. It also displays information about the Hello transmit interval and queue counts for unaccounted Hello acknowledgment.

Topology table Stores the route advertisements received from each neighbor. All routes in the AS are stored in the topology table, both successors, and feasible successors.

Route table Stores the routes that are currently in use to make local routing decisions. Anything in the routing table is considered a successor route.

We’ll explore more of EIGRP’s features in greater detail soon, beginning with a look at the metrics associated with particular routes. After that, I’ll cover the decision-making process that’s used to select the best routes, and then we’ll review the procedures followed when routes change.

Configuring EIGRP

I know what you’re thinking! “We’re going to jump in to configuring EIGRP already when I’ve heard how complex it is?” No worries here—what I’m about to show is basic, and I know you won’t have a problem with it at all! We’re going to start with the easy part of EIGRP, and by configuring it on our little internetwork, you’ll learn a lot more this way than you would if I just continued explaining more at this point. After we’ve completed the initial configuration, we’ll fine-tune it and have fun experimenting with it throughout this chapter!

But first, keep in mind that even though EIGRP can be configured for IP, IPv6, IPX, and AppleTalk, as a future Cisco Certified Network Associate R/S your focus needs to zero in on configuring EIGRP for IP and IPv6.

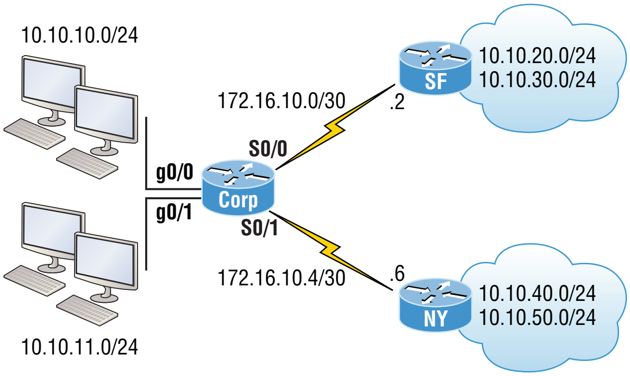
Okay, there are two modes for entering EIGRP commands: router configuration mode and interface configuration mode. In router configuration mode, we’ll enable the protocol, determine which networks will run EIGRP, and set global factors. When in interface configuration mode, we’ll customize summaries and bandwidth.

To initiate an EIGRP session on a router, I'll use the `router eigrp` command followed by our network's AS number. After that, we'll enter the specific numbers of the networks that we want to connect to the router using the `network` command followed by the network number. This is pretty straightforward stuff—if you can configure RIP, then you can configure EIGRP!

Just so you know, we'll use the same network I used in the previous CCENT routing chapters, but I'm going to connect more networks so we can look deeper into EIGRP. With that, I'm going to enable EIGRP for autonomous system 20 on our Corp router connected to four networks.

Figure 5.5 shows the network we'll be configuring throughout this chapter and the next chapter. Here's the Corp configuration:

FIGURE 5.5 Configuring our little internetwork with EIGRP



```
Corp#config t
Corp(config)#router eigrp 20
Corp(config-router)#network 172.16.0.0
Corp(config-router)#network 10.0.0.0
```

Remember, just as we would when configuring RIP, we need to use the classful network address, which is all subnet and host bits turned off. This is another thing that makes EIGRP so great: it has the complexity of a link-state protocol running in the background and the same easy configuration process used for RIP!

NOTE Understand that the AS number is irrelevant—that is, as long as all routers use the same number! You can use any number from 1 to 65,535.

But wait, the EIGRP configuration can't be that easy, can it? A few simple EIGRP commands and my network just works? Well, it can be and usually is, but not always.

Remember the wildcards you learned about in your OSPF and access list configurations in your preparation for the Cisco exam? We can go with the same type of configuration as we did with OSPF. Let's say, for example, that we wanted to advertise all the directly connected networks with EIGRP off the Corp router. By using the command `network 10.0.0.0`, we can effectively advertise to all subnets within that classful network; however, take a look at this configuration now:

```
Corp#config t
Corp(config)#router eigrp 20
Corp(config-router)#network 10.10.11.0 0.0.0.255
Corp(config-router)#network 172.16.10.0 0.0.0.3
Corp(config-router)#network 172.16.10.4 0.0.0.3
```

This configuration should look pretty familiar to you because by now you should have a solid understanding of how wildcards are configured. This configuration will advertise the network connected to g0/1 on the Corp router as well as the two WAN links. Still, all we accomplished with this configuration was to stop the g0/0 interface from being placed into the EIGRP process, and unless you have tens of thousands of networks worldwide, then there is really no need to use wildcards because they don't provide any other administrative purpose other than what I've already described.

Now let's take a look at the simple configuration needed for the SF and NY routers in our internetwork:

```
SF(config)#router eigrp 20
SF(config-router)#network 172.16.0.0
SF(config-router)#network 10.0.0.0
000060:%DUAL-5-NBRCHANGE:IP-EIGRP(0) 20:Neighbor 172.16.10.1 (Serial0/0/0) is up:
new adjacency


NY(config)#router eigrp 20
NY(config-router)#network 172.16.0.0
NY(config-router)#network 10.0.0.0
*Jun 26 02:41:36:%DUAL-5-NBRCHANGE:IP-EIGRP(0) 20:Neighbor 172.16.10.5
(Serial0/0/1) is up: new adjacency
```

Nice and easy—or is it? We can see that the SF and NY router created an adjacency to the Corp router, but are they actually sharing routing information? To find out, let's take a look at the number that I pointed out as the autonomous system (AS) number in the configuration.

EIGRP uses ASs to identify the group of routers that will share route information. Only routers that have the same AS share routes. The range of values we can use to create an AS with EIGRP is 1–65535:

```
Corp(config)#router eigrp ?
  <1-65535>  Autonomous System
  WORD       EIGRP Virtual-Instance Name
Corp(config)#router eigrp 20
```

Notice that I could have used any number from 1 to 65,535, but I chose to use 20 because it just felt good at the time. As long as all routers use the same number, they'll create an adjacency. Okay, now the AS makes sense, but it looks like I can type a `word` in the place of the AS number, and I can! Let's take a look at the configuration:

```
Corp(config)#router eigrp Todd
Corp(config-router)#address-family ipv4 autonomous-system 20
Corp(config-router-af)#network 10.0.0.0
Corp(config-router-af)#network 172.16.0.0
```

What I just showed you is not part of the Cisco exam objectives, but it's also not really necessary for any IPv4 routing configuration in your network. The previous configuration examples I've gone through so far in this chapter covers the objectives and works just fine, but I included this last configuration example because it's now an option in IOS 15.0 code.

VLSM Support and Summarization

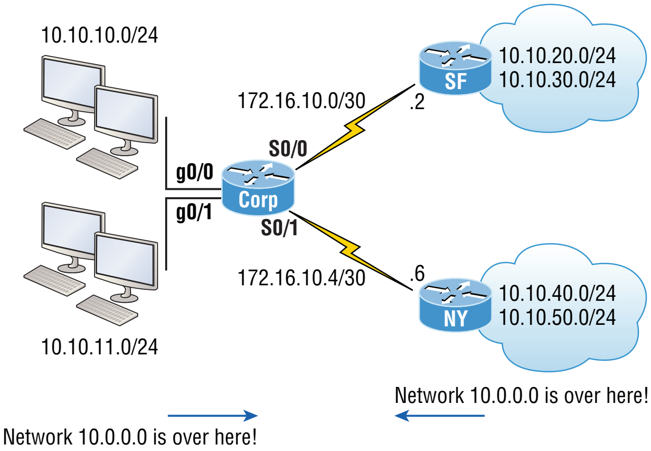
Being one of the more sophisticated classless routing protocols, EIGRP supports using variable length subnet masks. This is good because it allows us to conserve address space by using subnet masks that map to specific host requirements in a much better way. Being able to use 30-bit subnet masks for the point-to-point networks that I configured in our internetwork is a great example. Plus, because the subnet mask is propagated with every route update, EIGRP also supports the use of discontiguous subnets, giving us greater administrative flexibility when designing a network IP address scheme. Another versatile feature is that EIGRP allows us to use and place route summaries at strategically optimal locations throughout the EIGRP network to reduce the size of the route table.

Keep in mind that EIGRP automatically summarizes networks at their classful boundaries and supports the manual creation of summaries at any and all EIGRP routers. This is usually a good thing, but by checking out the routing table in the Corp router, you can see the possible complications that auto-summarization can cause:

```
Corp#sh ip route
[output cut]
     172.16.0.0/16 is variably subnetted, 3 subnets, 2 masks
C       172.16.10.4/30 is directly connected, Serial0/1
C       172.16.10.0/30 is directly connected, Serial0/0
D       172.16.0.0/16 is a summary, 00:01:37, Null0
     10.0.0.0/8 is variably subnetted, 3 subnets, 2 masks
C       10.10.10.0/24 is directly connected, GigabitEthernet0/0
D       10.0.0.0/8 is a summary, 00:01:19, Null0
C       10.10.11.0/24 is directly connected, GigabitEthernet0/1
```

Now this just doesn't look so good—both 172.16.0.0 and 10.0.0.0/8 are being advertised as summary routes injected by EIGRP, but we have multiple subnets in the 10.0.0.0/8 classful network address, so how would the Corp router know how to route to a specific network like 10.10.20.0? The answer is, it wouldn't. Let's see why in Figure 5.6.

FIGURE 5.6 Discontiguous networks


10.10.10.0/24
10.10.11.0/24
g0/0
g0/1
Corp
S0/0
S0/1
172.16.10.0/30
172.16.10.4/30
.2
.6
SF
NY
10.10.20.0/24
10.10.30.0/24
10.10.40.0/24
10.10.50.0/24
Network 10.0.0.0 is over here!
Network 10.0.0.0 is over here!

The networks we're using are considered a discontiguous network because we have the 10.0.0.0/8 network subnetted across a different class of address; the 172.16.0.0 network, with 10.0.0.0/8 subnets on both sides of the WAN links.

You can see that the SF and NY routers will both create an automatic summary of 10.0.0.0/8 and then inject it into their routing tables. This is a common problem, and an important one that Cisco really wants you to understand (by including it in the objectives)! With this type of topology, disabling automatic summarization is definitely the better option. Actually, it's the only option if we want this network to work.

Let's take a look at the routing tables on the NY and SF routers to find out what they're seeing:

```
SF>sh ip route
[output cut]
     172.16.0.0/16 is variably subnetted, 3 subnets, 3 masks
C       172.16.10.0/30 is directly connected, Serial0/0/0
D       172.16.10.0/24 [90/2681856] via 172.16.10.1, 00:54:58, Serial0/0/0
D       172.16.0.0/16 is a summary, 00:55:12, Null0
     10.0.0.0/8 is variably subnetted, 3 subnets, 2 masks
D       10.0.0.0/8 is a summary, 00:54:58, Null0
C       10.10.20.0/24 is directly connected, FastEthernet0/0
C       10.10.30.0/24 is directly connected, Loopback0
SF>
```

```
NY>sh ip route
[output cut]
     172.16.0.0/16 is variably subnetted, 2 subnets, 2 masks
C       172.16.10.4/30 is directly connected, Serial0/0/1
D       172.16.0.0/16 is a summary, 00:55:56, Null0
     10.0.0.0/8 is variably subnetted, 3 subnets, 2 masks
D       10.0.0.0/8 is a summary, 00:55:26, Null0
C       10.10.40.0/24 is directly connected, FastEthernet0/0
C       10.10.50.0/24 is directly connected, Loopback0
NY>ping 10.10.10.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.10.10.1, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
NY>
```

The confirmed answer is that our network isn't working because we're discontiguous and our classful boundaries are auto-summarizing. We can see that EIGRP is injecting summary routes into both the SF and NY routing tables.

We need to advertise our subnets in order to make this work, and here's how we make that happen, starting with the Corp router:

```
Corp#config t
Corp(config)#router eigrp 20
Corp(config-router)#no auto-summary
Corp(config-router)#
*Feb 25 18:29:30%DUAL-5-NBRCHANGE:IP-EIGRP(0) 20:Neighbor 172.16.10.6 (Serial0/1)
 is resync: summary configured
*Feb 25 18:29:30%DUAL-5-NBRCHANGE:IP-EIGRP(0) 20:Neighbor 172.16.10.2 (Serial0/0)
 is resync: summary configured
Corp(config-router)#
```

Okay—our network still isn't working because the other routers are still sending a summary. So let's configure the SF and NY routers to advertise subnets:

```
SF#config t
SF(config)#router eigrp 20
SF(config-router)#no auto-summary
SF(config-router)#
000090:%DUAL-5-NBRCHANGE:IP-EIGRP(0) 20:Neighbor 172.16.10.1 (Serial0/0/0) is
resync: summary configured
```

```
NY#config t
NY(config)#router eigrp 20
NY(config-router)#no auto-summary
NY(config-router)#
*Jun 26 21:31:08%DUAL-5-NBRCHANGE:IP-EIGRP(0) 20:Neighbor 172.16.10.5
(Serial0/0/1)
is resync: summary configured
```

Let's take a look at the Corp router's output now:

```
Corp(config-router)#do show ip route
[output cut]
     172.16.0.0/30 is subnetted, 2 subnets
C       172.16.10.4 is directly connected, Serial0/1
C       172.16.10.0 is directly connected, Serial0/0
     10.0.0.0/24 is subnetted, 6 subnets
C       10.10.10.0 is directly connected, GigabitEthernet0/0
C       10.10.11.0 is directly connected, GigabitEthernet0/1
D       10.10.20.0 [90/3200000] via 172.16.10.2, 00:00:27, Serial0/0
D       10.10.30.0 [90/3200000] via 172.16.10.2, 00:00:27, Serial0/0
D       10.10.40.0 [90/2297856] via 172.16.10.6, 00:00:29, Serial0/1
D       10.10.50.0 [90/2297856] via 172.16.10.6, 00:00:30, Serial0/1
Corp# ping 10.10.20.1


Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.10.20.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms
```

Wow, what a difference compared to the previous routing table output! We can see all the subnets now. It would be hard to justify using auto-summarization today. If you want to summarize, it should definitely be done manually. Always typing in `no auto-summary` under RIPv2 and EIGRP is common practice today.

> The new 15.*x* code auto-summarization feature is disabled by default, as it should be. But don't think that discontiguous networks and disabling auto-summary are no longer topics in the Cisco exam objectives, because they most certainly are!

Controlling EIGRP Traffic

But what if you need to stop EIGRP from working on a specific interface? Maybe it's a connection to your ISP, or where we didn't want to have the g0/0 interface be part of the

EIGRP process as in our earlier example. All you need to do is to flag the interface as passive, and to do this from an EIGRP session, just use this command:

```
passive-interface interface-type interface-number
```

This works because the *`interface-type`* portion defines the type of interface and the *`interface-number`* portion defines the number of the interface. The following command makes interface serial 0/0 into a passive interface:

```
Corp(config)#router eigrp 20
Corp(config-router)#passive-interface g0/0
```

What we've accomplished here is to prevent this interface from sending or reading received Hello packets so that it will no longer form adjacencies or send or receive route information. But this still won't stop EIGRP from advertising the subnet of this interface out all other interfaces without using wildcards. This really illustrates the reason you must understand why and when to use wildcards, as well as, what the `passive-interface` command does. This knowledge really helps you to make an informed decision on which command you need to use to meet your specific business requirements!

> The impact of the `passive-interface` command depends upon the routing protocol under which the command is issued. For example, on an interface running RIP, the `passive-interface` command will prohibit sending route updates but will permit receiving them. An RIP router with a passive interface will still learn about the networks advertised by other routers. This is different from EIGRP, where an interface configured with the `passive-interface` command will neither send nor read received Hellos.

Typically, EIGRP neighbors use multicast to exchange routing updates. You can change this by specifically telling the router about a particular neighbor, which will ensure that unicast packets will only be used for the routing updates with that specific neighbor. To take advantage of this feature, apply the `neighbor` command and execute it under the EIGRP process.

I'm going to configure the Corp router with information about routers SF and NY:

```
Corp(config)#router eigrp 20
Corp(config-router)#neighbor 172.16.10.2
Corp(config-router)#neighbor 172.16.10.6
```

Understand that you don't need to use the preceding commands to create neighbor relationships, but they're available if you need them.

EIGRP Metrics

Unlike many other protocols that use a single element to compare routes and select the best possible path, EIGRP uses a combination of these four factors:

- *Bandwidth*
- *Delay*

- *Load*
- *Reliability*

It's worth noting that there's a fifth element, *maximum transmission unit (MTU)*, which has never been used in EIGRP metrics calculations though it's still a required parameter in some EIGRP-related commands—especially those involving redistribution. The value of the MTU element represents the smallest MTU value encountered along the path to the destination network.

Also good to know is that there's a mathematical formula that combines the four main elements to create a single value representing just how good a given route actually is. The higher the metric associated with it, the less desirable the route. Here's that formula:

$$metric = [K_1 \times Bandwidth + (K_2 \times Bandwidth) / (256 - Load) + K_3 \times Delay] \times [K_5 / (Reliability + K_4)]$$

The formula's components break down like this:

- By default, $K_1 = 1$, $K_2 = 0$, $K_3 = 1$, $K_4 = 0$, $K_5 = 0$.
- *Delay* equals the sum of all the delays of the links along the path.
  - *Delay* = [Delay in 10s of microseconds] × 256.
- *Bandwidth* is the lowest bandwidth of the links along the path.
  - *Bandwidth* = [10000000 / (bandwidth in Kbps)] × 256.
- By default, *metric* = lowest bandwidth along path + sum of all delays along path.

If necessary, you can adjust the constant K values on a per-interface basis, but I would recommend that you only do this under the direction of the Cisco Technical Assistance Center (TAC). Metrics are tuned to change the manner in which routes are calculated. The K values can be seen with a `show ip protocols` output:

```
Corp#sh ip protocols
*** IP Routing is NSF aware ***


Routing Protocol is "eigrp 1"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Default networks flagged in outgoing updates
  Default networks accepted from incoming updates
  EIGRP-IPv4 Protocol for AS(1)
    Metric weight K1=1, K2=0, K3=1, K4=0, K5=0
```

Notice that that the K1 and K3 values are enabled by default—for example, K1 = 1. Table 5.1 shows the relationship between each constant and the metric it affects.

TABLE 5.1 Metric association of K values

Constant	Metric
K1	Bandwidth (B_e)
K2	Load (utilization on path)
K3	Delay (D_c)
K4	Reliability (r)
K5	MTU

Each constant is used to assign a weight to a specific variable, meaning that when the metric is calculated, the algorithm will assign a greater importance to the specified metric. This is very cool because it means that by assigning a weight, you get to specify the factor that's most important to you. For example, if bandwidth is your priority, you would assign K1 to weight it accordingly, but if delay is totally unacceptable, then K3 would be assigned a greater weight. A word of caution though: Always remember that any changes to the default values could result in instability and convergence problems, particularly if delay or reliability values are constantly changing! But if you're looking for something to do on a rainy Saturday, it's an interesting experiment to pass some time and gain some nice networking insight!

Maximum Paths and Hop Count

By default, EIGRP can provide equal-cost load balancing across up to 4 links. RIP and OSPF do this too. But you can have EIGRP actually load-balance across up to 32 links with 15.0 code (equal or unequal) by using the following command:

```
Corp(config)#router eigrp 10
Corp(config-router)#maximum-paths ?
  <1-32>  Number of paths
```

As I mentioned, pre–15.0 code routers allowed up to 16 paths to remote networks, which is still a lot!

EIGRP has a default maximum hop count of 100 for route update packets, but it can be set up to 255. Chances are you wouldn't want to ever change this, but if you did, here is how you would do it:

```
Corp(config)#router eigrp 10
Corp(config-router)#metric maximum-hops ?
  <1-255>  Hop count
```

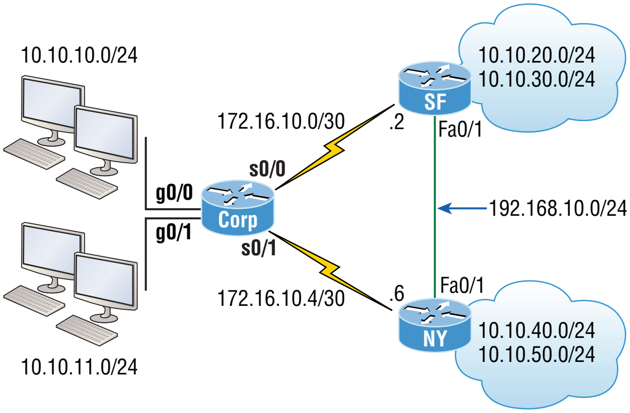
As you can see from this router output, EIGRP can be set to a maximum of 255 hops. Even though it doesn't use hop count in the path metric calculation, it still uses the maximum hop count to limit the scope of the AS.

Route Selection

Now that you've got a good idea how EIGRP works and also how easy it actually is to configure, it's probably clear that determining the best path simply comes down to seeing which one gets awarded the lowest metric. But it's not the winning path that really sets EIGRP apart from other protocols. You know that EIGRP stores route information from its neighbors in its topology table and that as long as a given neighbor remains alive, it will rarely throw out anything it has learned from that neighbor. This makes EIGRP able to flag the best routes in its topology table for positioning in its local routing table, enabling it to flag the next-best routes as alternatives if the best route goes down.

In Figure 5.7 you can see that I added another Fast Ethernet link between the SF and NY routers. This will give us a great opportunity to play with the topology and routing tables!

FIGURE 5.7 EIGRP route selection process



First, let's take another look at the routing table on the Corp router before I bring up the new interfaces:

```
172.16.0.0/30 is subnetted, 2 subnets
C       172.16.10.4 is directly connected, Serial0/1
C       172.16.10.0 is directly connected, Serial0/0
     10.0.0.0/24 is subnetted, 6 subnets
C       10.10.10.0 is directly connected, GigabitEthernet0/0
C       10.10.11.0 is directly connected, GigabitEthernet0/1
D       10.10.20.0 [90/3200000] via 172.16.10.2, 00:00:27, Serial0/0
D       10.10.30.0 [90/3200000] via 172.16.10.2, 00:00:27, Serial0/0
```

```
D       10.10.40.0 [90/2297856] via 172.16.10.6, 00:00:29, Serial0/1
D       10.10.50.0 [90/2297856] via 172.16.10.6, 00:00:30, Serial0/1
```

We can see the three directly connected interfaces, as well as, the other four networks injected into the routing table by EIGRP. Now, I'll add the network 192.168.10.0/24 between the SF and NY routers, then enable the interfaces.

And let's check out the routing table of the Corp router now that I've configured that link:

```
D    192.168.10.0/24 [90/2172416] via 172.16.10.6, 00:04:27, Serial0/1
 172.16.0.0/30 is subnetted, 2 subnets
C       172.16.10.4 is directly connected, Serial0/1
C       172.16.10.0 is directly connected, Serial0/0
     10.0.0.0/24 is subnetted, 6 subnets
C       10.10.10.0 is directly connected, GigabitEthernet0/0
C       10.10.11.0 is directly connected, GigabitEthernet0/1
D       10.10.20.0 [90/3200000] via 172.16.10.2, 00:00:27, Serial0/0
D       10.10.30.0 [90/3200000] via 172.16.10.2, 00:00:27, Serial0/0
D       10.10.40.0 [90/2297856] via 172.16.10.6, 00:00:29, Serial0/1
D       10.10.50.0 [90/2297856] via 172.16.10.6, 00:00:30, Serial0/1
```

Okay—that's weird. The only thing different I see is one path to the 192.168.10.0/24 network listed first. Glad it is there, which means that we can route to that network. Notice that we can reach the network from the Serial0/1 interface, but what happened to my link to the SF router—shouldn't we have an advertisement from that router and be load-balancing? Let's take a look the topology table to find out what's going on:

```
Corp#sh ip eigrp topology
IP-EIGRP Topology Table for AS(20)/ID(10.10.11.1)


Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply,
       r - reply Status, s - sia Status


P 10.10.10.0/24, 1 successors, FD is 128256
        via Connected, GigbitEthernet0/0
P 10.10.11.0/24, 1 successors, FD is 128256
        via Connected, GigbitEthernet0/1
P 10.10.20.0/24, 1 successors, FD is 2300416
        via 172.16.10.6 (2300416/156160), Serial0/1
        via 172.16.10.2 (3200000/128256), Serial0/0
P 10.10.30.0/24, 1 successors, FD is 2300416
        via 172.16.10.6 (2300416/156160), Serial0/1
        via 172.16.10.2 (3200000/128256), Serial0/0
```

```
P 10.10.40.0/24, 1 successors, FD is 2297856
        via 172.16.10.6 (2297856/128256), Serial0/1
        via 172.16.10.2 (3202560/156160), Serial0/0
P 10.10.50.0/24, 1 successors, FD is 2297856
        via 172.16.10.6 (2297856/128256), Serial0/1
        via 172.16.10.2 (3202560/156160), Serial0/0
P 192.168.10.0/24, 1 successors, FD is 2172416
        via 172.16.10.6 (2172416/28160), Serial0/1
        via 172.16.10.2 (3074560/28160), Serial0/0
P 172.16.10.4/30, 1 successors, FD is 2169856
        via Connected, Serial0/1
P 172.16.10.0/30, 1 successors, FD is 3072000
        via Connected, Serial0/0
```

Okay, we can see there are two paths to the 192.168.10.0/24 network, but it's using the next hop of 172.16.10.6 (NY) because the feasible distance (FD) is less! The advertised distance from both routers is 28160, but the cost to get to each router via the WAN links is the same. This means the FD is not the same, meaning we're not load-balancing by default.

Both WAN links are a T1, so this should have load-balanced by default, but EIGRP has determined that it costs more to go through SF than through NY. Since EIGRP uses bandwidth and delay of the line to determine the best path, we can use the `show interfaces` command to verify our stats like this:

```
Corp#sh int s0/0
Serial0/0 is up, line protocol is up
  Hardware is PowerQUICC Serial
  Description: <<Connection to CR1>>
  Internet address is 172.16.10.1/30
  MTU 1500 bytes, BW 1000 Kbit, DLY 20000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation HDLC, loopback not set Keepalive set (10 sec)


Corp#sh int s0/1
Serial0/1 is up, line protocol is up
  Hardware is PowerQUICC Serial
  Internet address is 172.16.10.5/30
  MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation HDLC, loopback not set Keepalive set (10 sec)
```

I highlighted the statistic that EIGRP uses to determine the metrics to a next-hop router: MTU, bandwidth, delay, reliability, and load, with bandwidth and delay enabled by default.

We can see that the bandwidth on the Serial0/0 interface is set to 1000 Kbit, which is not the default bandwidth. Serial0/1 is set to the default bandwidth of 1544 Kbit.

Let's set the bandwidth back to the default on the s0/0 interface and we should start load-balancing to the 192.168.10.0 network. I'll just use the `no bandwidth` command, which will set it back to its default of 1544 Mbps:

```
Corp#config t
Corp(config)#int s0/0
Corp(config-if)#no bandwidth
Corp(config-if)#^Z
```

Now let's take a look at the topology table and see if we're equal.

```
Corp#sh ip eigrp topo | section 192.168.10.0
P 192.168.10.0/24, 2 successors, FD is 2172416
        via 172.16.10.2 (2172416/28160), Serial0/0
        via 172.16.10.6 (2172416/28160), Serial0/1
```

Since the topology tables can get really huge in most networks, the `show ip eigrp topology | section` *`network`* command comes in handy because it allows us to see information about the network we want to look into in a couple of lines.

Let's use the `show ip route` *`network`* command and check out what is going on there:

```
Corp#sh ip route 192.168.10.0
Routing entry for 192.168.10.0/24
  Known via "eigrp 20", distance 90, metric 2172416, type internal
  Redistributing via eigrp 20
  Last update from 172.16.10.2 on Serial0/0, 00:05:18 ago
  Routing Descriptor Blocks:
  * 172.16.10.6, from 172.16.10.6, 00:05:18 ago, via Serial0/1
      Route metric is 2172416, traffic share count is 1
      Total delay is 20100 microseconds, minimum bandwidth is 1544 Kbit
      Reliability 255/255, minimum MTU 1500 bytes
      Loading 1/255, Hops 1
    172.16.10.2, from 172.16.10.2, 00:05:18 ago, via Serial0/0
      Route metric is 2172416, traffic share count is 1
      Total delay is 20100 microseconds, minimum bandwidth is 1544 Kbit
      Reliability 255/255, minimum MTU 1500 bytes
      Loading 1/255, Hops 1
```

Lots of detail about our routes to the 192.168.10.0 network! The Corp route has two equal-cost links to the 192.168.10.0 network. And to reveal load balancing even better, we'll just use the plain, ever useful `show ip route` command:

```
Corp#sh ip route
[output cut]
```

```
D    192.168.10.0/24 [90/2172416] via 172.16.10.6, 00:05:35, Serial0/1
                     [90/2172416] via 172.16.10.2, 00:05:35, Serial0/0
```

Now we can see that there are two successor routes to the 192.168.10.0 network. Pretty sweet! But in the routing table, there's one path to 192.168.20.0 and 192.168.30.0, with the link between the SF and NY routers being feasible successors. And it's the same with the 192.168.40.0 and 192.168.50.0 networks. Let's take a look at the topology table to examine this more closely:

```
Corp#sh ip eigrp topology
IP-EIGRP Topology Table for AS(20)/ID(10.10.11.1)


Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply,
       r - reply Status, s - sia Status


P 10.10.10.0/24, 1 successors, FD is 128256
        via Connected, GigabitEthernet0/0
P 10.10.11.0/24, 1 successors, FD is 128256
        via Connected, GigabitEthernet0/1
P 10.10.20.0/24, 1 successors, FD is 2297856
        via 172.16.10.2 (2297856/128256), Serial0/0
        via 172.16.10.6 (2300416/156160), Serial0/1
P 10.10.30.0/24, 1 successors, FD is 2297856
        via 172.16.10.2 (2297856/128256), Serial0/0
        via 172.16.10.6 (2300416/156160), Serial0/1
P 10.10.40.0/24, 1 successors, FD is 2297856
        via 172.16.10.6 (2297856/128256), Serial0/1
        via 172.16.10.2 (2300416/156160), Serial0/0
P 10.10.50.0/24, 1 successors, FD is 2297856
        via 172.16.10.6 (2297856/128256), Serial0/1
        via 172.16.10.2 (2300416/156160), Serial0/0
P 192.168.10.0/24, 2 successors, FD is 2172416
        via 172.16.10.2 (2172416/28160), Serial0/0
        via 172.16.10.6 (2172416/28160), Serial0/1
P 172.16.10.4/30, 1 successors, FD is 2169856
        via Connected, Serial0/1
P 172.16.10.0/30, 1 successors, FD is 2169856
        via Connected, Serial0/0
```

It is nice that we can see that we have a successor and a feasible successor to each network, so we know that EIGRP is doing its job. Let's take a close look at the links to 10.10.20.0 now and dissect what it's telling us:

```
P 10.10.20.0/24, 1 successors, FD is 2297856
        via 172.16.10.2 (2297856/128256), Serial0/0
        via 172.16.10.6 (2300416/156160), Serial0/1
```

Okay—first, we can see that it's passive (P), which means that it has found all the usable paths to the network 10.10.20.0 and is happy! If we see active (A), that means that EIGRP is not happy at all and is querying its neighbors for a new path to that network. The `(2297856/128256)` is the FD/AD, meaning that the SF router is advertising the 10.10.20.0 network as a cost of 128256, which is the AD. The Corp router adds the bandwidth and delay of the line to get to the SF router and then adds that number to the AD (128256) to come up with a total cost (FD) of 2297856 to get to network 10.10.20.0.

WARNING

To become a CCNA R/S, you must understand how to read a topology table!

Unequal-Cost Load Balancing

As with all routing protocols running on Cisco routers, EIGRP automatically supports load balancing over four equal-cost routes and can be configured to support up to 32 equal-cost paths with IOS 15.0 code. As you know, previous IOS versions supported up to 16. I've mentioned this a few times in this chapter already, but I want to show you how to configure unequal-cost load balancing with EIGRP. First let's take a look at the Corp router by typing in the `show ip protocols` command:

```
Corp#sh ip protocols
Routing Protocol is "eigrp 20"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Default networks flagged in outgoing updates
  Default networks accepted from incoming updates
  EIGRP metric weight K1=1, K2=0, K3=1, K4=0, K5=0
  EIGRP maximum hopcount 100
  EIGRP maximum metric variance 1
  Redistributing: eigrp 20
  EIGRP NSF-aware route hold timer is 240s
  Automatic network summarization is not in effect
  Maximum path: 4
  Routing for Networks:
    10.0.0.0
    172.16.0.0
  Routing Information Sources:
    Gateway         Distance      Last Update
    (this router)         90      19:15:10
    172.16.10.6           90      00:25:38
    172.16.10.2           90      00:25:38
  Distance: internal 90 external 170
```

The `variance 1` means equal path load balancing with the maximum paths set to 4 by default. Unlike most other protocols, EIGRP also supports unequal-cost load balancing through the use of the `variance` parameter.

To clarify, let's say the parameter has been set to a variance of 2. This would effectively load-balance traffic across the best route plus any route with a feasible distance of up to twice as large. But still keep in mind that load balancing occurs in proportion with and relative to the cost of the route, meaning that more traffic would travel across the best route than the suboptimal one.

Let's configure the variance on the Corp router and see if we can load-balance across our feasible successors now:

```
Corp# config t
Corp(config)#router eigrp 20
Corp(config-router)#variance 2
Corp(config-router)#
*Feb 26 22:24:24:IP-EIGRP(Default-IP-Routing-Table:20):route installed for
10.10.20.0
*Feb 26 22:24:24:IP-EIGRP(Default-IP-Routing-Table:20):route installed for
10.10.20.0
*Feb 26 22:24:24:IP-EIGRP(Default-IP-Routing-Table:20):route installed for
10.10.30.0
*Feb 26 22:24:24:IP-EIGRP(Default-IP-Routing-Table:20):route installed for
10.10.30.0
*Feb 26 22:24:24:IP-EIGRP(Default-IP-Routing-Table:20):route installed for
10.10.40.0
*Feb 26 22:24:24:IP-EIGRP(Default-IP-Routing-Table:20):route installed for
10.10.40.0
*Feb 26 22:24:24:IP-EIGRP(Default-IP-Routing-Table:20):route installed for
10.10.50.0
*Feb 26 22:24:24:IP-EIGRP(Default-IP-Routing-Table:20):route installed for
10.10.50.0
*Feb 26 22:24:24:IP-EIGRP(Default-IP-Routing-Table:20):route installed for
192.168.10.0
*Feb 26 22:24:24:IP-EIGRP(Default-IP-Routing-Table:20):route installed for
192.168.10.0
```

```
Corp(config-router)#do show ip route
[output cut]
D    192.168.10.0/24 [90/2172416] via 172.16.10.6, 00:00:18, Serial0/1
                     [90/2172416] via 172.16.10.2, 00:00:18, Serial0/0
     172.16.0.0/30 is subnetted, 2 subnets
C       172.16.10.4 is directly connected, Serial0/1
C       172.16.10.0 is directly connected, Serial0/0
     10.0.0.0/24 is subnetted, 6 subnets
C       10.10.10.0 is directly connected, GigabitEthernet0/0
C       10.10.11.0 is directly connected, GigabitEthernet0/1
D       10.10.20.0 [90/2300416] via 172.16.10.6, 00:00:18, Serial0/1
                   [90/2297856] via 172.16.10.2, 00:00:19, Serial0/0
D       10.10.30.0 [90/2300416] via 172.16.10.6, 00:00:19, Serial0/1
                   [90/2297856] via 172.16.10.2, 00:00:19, Serial0/0
D       10.10.40.0 [90/2297856] via 172.16.10.6, 00:00:19, Serial0/1
                   [90/2300416] via 172.16.10.2, 00:00:19, Serial0/0
D       10.10.50.0 [90/2297856] via 172.16.10.6, 00:00:20, Serial0/1
                   [90/2300416] via 172.16.10.2, 00:00:20, Serial0/0
Corp(config-router)#
```

Nice—it worked! Now we have two paths to each remote network in the routing table, even though the feasible distances to each route aren't equal. Don't forget that unequal load balancing is not enabled by default and that you can perform load-balancing through paths that have up to 128 times worse metrics than the successor route!

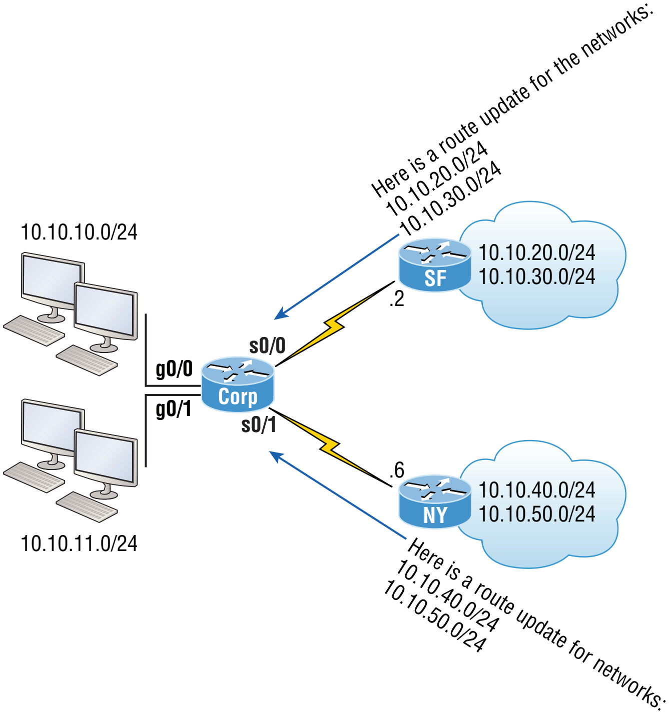
Split Horizon

Split horizon is enabled on interfaces by default, which means that if a route update is received on an interface from a neighbor router, this interface will not advertise those networks back out to the neighbor router who sent them. Let's take a look at an interface and then go through an example:

```
Corp#sh ip int s0/0
Serial0/0 is up, line protocol is up
  Internet address is 172.16.10.1/24
  Broadcast address is 255.255.255.255
  Address determined by setup command
  MTU is 1500 bytes
  Helper address is not set
  Directed broadcast forwarding is disabled
  Multicast reserved groups joined: 224.0.0.10
  Outgoing access list is not set
  Inbound  access list is not set
  Proxy ARP is enabled
  Local Proxy ARP is disabled
  Security level is default
  Split horizon is enabled
[output cut]
```

Okay—we can see that split horizon is enabled by default. But what does this really mean? Most of the time it's more helpful than harmful, but let's check out our internetwork in Figure 5.8 so I can really explain what split horizon is doing.

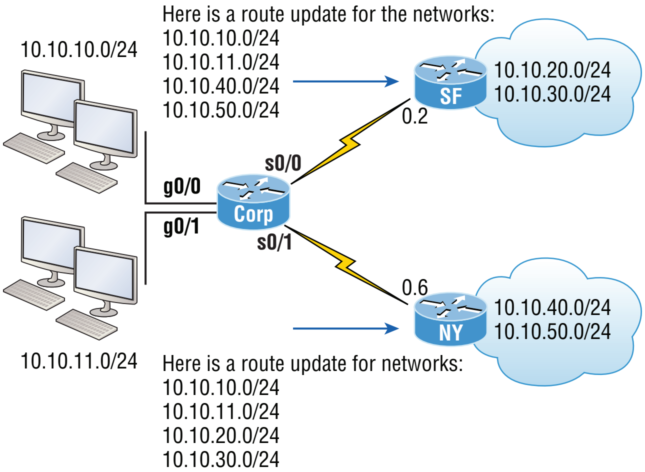
FIGURE 5.8 Split horizon in action, part 1


Here is a route update for the networks:
10.10.20.0/24
10.10.30.0/24
10.10.10.0/24
10.10.20.0/24
10.10.30.0/24
SF
.2
s0/0
g0/0
Corp
g0/1
s0/1
.6
10.10.40.0/24
10.10.50.0/24
NY
10.10.11.0/24
Here is a route update for networks:
10.10.40.0/24
10.10.50.0/24

Notice that the SF and NY routers are each advertising their routes to the Corp router. Now, let's see what the Corp router sends back to each router in Figure 5.9.

Can you see that the Corp router is not advertising back out the advertised networks that it received on each interface? This is saving the SF and NY routers from receiving the incorrect route information that they could possibly get to their own network through the Corp router, which we know is wrong.

FIGURE 5.9 Split horizon in action, part 2


Here is a route update for the networks:
10.10.10.0/24
10.10.11.0/24
10.10.40.0/24
10.10.50.0/24
10.10.10.0/24
SF
10.10.20.0/24
10.10.30.0/24
0.2
s0/0
g0/0
Corp
g0/1
s0/1
0.6
NY
10.10.40.0/24
10.10.50.0/24
10.10.11.0/24
Here is a route update for networks:
10.10.10.0/24
10.10.11.0/24
10.10.20.0/24
10.10.30.0/24

So how can this cause a problem? After all, it seems reasonable not to send misinformation back to an originating router, right? You'll see this create a problem on point-to-multipoint links, such as Frame Relay, when multiple remote routers connect to a single interface at the Corp location. We can use logical interfaces, called subinterfaces, which I'll tell you all about in Chapter 7, "Wide Area Networks," to solve the split horizon issue on a point-to-multipoint interface.

Visit ccna .gg/ch5/a for a companion MicroNugget from CBT Nuggets.

Verifying and Troubleshooting EIGRP

Even though EIGRP usually runs smoothly and is relatively low maintenance, there are several commands you need to memorize for using on a router that can be super helpful when troubleshooting EIGRP! I've already showed you a few of them, but I'm going to demonstrate all the tools you'll need to verify and troubleshoot EIGRP now. Table 5.2 contains all of the commands you need to know for verifying that EIGRP is functioning well and offers a brief description of what each command does.

TABLE 5.2 EIGRP troubleshooting commands

Command	Description/Function
`show ip eigrp neighbors`	Shows all EIGRP neighbors
`show ip eigrp interfaces`	Lists the interfaces on which the router has actually enabled EIGRP
`show ip route eigrp`	Shows EIGRP entries in the routing table
`show ip eigrp topology`	Shows entries in the EIGRP topology table
`show ip eigrp traffic`	Shows the packet count for EIGRP packets sent and received
`show ip protocols`	Shows information about the active protocol sessions

When troubleshooting an EIGRP problem, it's always a good idea to start by getting an accurate map of the network, and the best way to do that is by using the `show ip eigrp neighbors` command to find out who your directly connected neighbors are. This command shows all adjacent routers that share route information within a given AS. If neighbors are missing, check the configuration, AS number, and link status on both routers to verify that the protocol has been configured correctly.

Let's execute the command on the Corp router:

```
Corp#sh ip eigrp neighbors
IP-EIGRP neighbors for process 20
H   Address                 Interface       Hold Uptime   SRTT   RTO  Q  Seq
                                            (sec)         (ms)       Cnt Num
1   172.16.10.2             Se0/0             11 03:54:25    1   200  0  127
0   172.16.10.6             Se0/1             11 04:14:47    1   200  0  2010
```

Here's a breakdown of the important information we can see in the preceding output:

- H indicates the order in which the neighbor was discovered.
- Hold time in seconds is how long this router will wait for a Hello packet to arrive from a specific neighbor.
- The Uptime value indicates how long the neighbor relationship has been established.
- The SRTT field is the smooth round-trip timer and represents how long it takes to complete a round-trip from this router to its neighbor and back. This value delimits how long to wait after a multicast for a reply from this neighbor. As mentioned earlier, the router will attempt to establish communication via unicasts if it doesn't receive a reply.
- The time between multicast attempts is specified by the Retransmission Time Out (RTO) field, which is based upon the SRTT values.

- The Q value tells us if there are any outstanding messages in the queue. We can make a mental note that there's a problem if we see consistently large values here!
- Finally, the Seq field shows the sequence number of the last update from that neighbor, which is used to maintain synchronization and avoid duplicate messages or their out-of-sequence processing.

The `neighbors` command is a great command, but we can get local status of our router by also using the `show ip eigrp interface` command like this:

```
Corp#sh ip eigrp interfaces
IP-EIGRP interfaces for process 20


                      Xmit Queue   Mean   Pacing Time   Multicast    Pending
Interface        Peers  Un/Reliable  SRTT   Un/Reliable   Flow Timer   Routes
Gi0/0              0        0/0         0       0/1           0            0
Se0/1              1        0/0         1       0/15         50            0
Se0/0              1        0/0         1       0/15         50            0
Gi0/1              0        0/0         0       0/1           0            0


Corp#sh ip eigrp interface detail s0/0
IP-EIGRP interfaces for process 20


                      Xmit Queue   Mean   Pacing Time   Multicast    Pending
Interface        Peers  Un/Reliable  SRTT   Un/Reliable   Flow Timer   Routes
Se0/0              1        0/0         1       0/15         50            0
  Hello interval is 5 sec
  Next xmit serial <none>
  Un/reliable mcasts: 0/0  Un/reliable ucasts: 21/26
  Mcast exceptions: 0  CR packets: 0  ACKs suppressed: 9
  Retransmissions sent: 0  Out-of-sequence rcvd: 0
  Authentication mode is not set
```

The first command, `show ip eigrp interfaces`, lists all interfaces for which EIGRP is enabled, as well as, those the router is currently sending Hello messages to in an attempt to find new EIGRP neighbors. The `show ip eigrp interface detail` *`interface`* command lists more details per interface, including the local router's own Hello interval. Understand that you can use these commands to verify that all your interfaces are within the AS process used by EIGRP, but also note that the passive interfaces won't show up in these outputs. So be sure to also check to see if an interface has been configured as passive if is not present in the outputs.

Okay, if all neighbors are present, then verify the routes learned. By executing the `show ip route eigrp` command, you're given a quick picture of the routes in the routing table.

If a certain route doesn't appear in the routing table, you need to verify its source. If the source is functioning properly, then check the topology table.

The routing table according to Corp looks like this:

```
D    192.168.10.0/24 [90/2172416] via 172.16.10.6, 02:29:09, Serial0/1
                     [90/2172416] via 172.16.10.2, 02:29:09, Serial0/0
     172.16.0.0/30 is subnetted, 2 subnets
C       172.16.10.4 is directly connected, Serial0/1
C       172.16.10.0 is directly connected, Serial0/0
     10.0.0.0/24 is subnetted, 6 subnets
C       10.10.10.0 is directly connected, Loopback0
C       10.10.11.0 is directly connected, Loopback1
D       10.10.20.0 [90/2300416] via 172.16.10.6, 02:29:09, Serial0/1
                   [90/2297856] via 172.16.10.2, 02:29:10, Serial0/0
D       10.10.30.0 [90/2300416] via 172.16.10.6, 02:29:10, Serial0/1
                   [90/2297856] via 172.16.10.2, 02:29:10, Serial0/0
D       10.10.40.0 [90/2297856] via 172.16.10.6, 02:29:10, Serial0/1
                   [90/2300416] via 172.16.10.2, 02:29:10, Serial0/0
D       10.10.50.0 [90/2297856] via 172.16.10.6, 02:29:11, Serial0/1
                   [90/2300416] via 172.16.10.2, 02:29:11, Serial0/0
```

You can see here that most EIGRP routes are referenced with a *D* and that their administrative distance is 90. Remember that the `[90/2300416]` represents AD/FD, and in the preceding output, EIGRP is performing equal- and unequal-cost load balancing between two links to our remote networks.

We can see this by looking closer at two different networks. Pay special attention to the FD of each output:

```
Corp#sh ip route | section 192.168.10.0
D    192.168.10.0/24 [90/2172416] via 172.16.10.6, 01:15:44, Serial0/1
                     [90/2172416] via 172.16.10.2, 01:15:44, Serial0/0
```

The preceding output show equal-cost load balancing, and here's our unequal-cost load balancing in action:

```
Corp#sh ip route | section 10.10.50.0
D       10.10.50.0 [90/2297856] via 172.16.10.6, 01:16:16, Serial0/1
                   [90/2300416] via 172.16.10.2, 01:16:16, Serial0/0
```

We can get the topology table displayed for us via the `show ip eigrp topology` command. If the route is in the topology table but not in the routing table, it's a pretty safe assumption that there's a problem between the topology database and the routing table. After all, there must be a good reason the topology database isn't adding the route into the routing table, right? We discussed this issue in detail earlier in the chapter, and it's oh so important!

Corp's topology table looks like this:

```
P 10.10.10.0/24, 1 successors, FD is 128256
        via Connected, GigabitEthernet0/0
P 10.10.11.0/24, 1 successors, FD is 128256
        via Connected, GigabitEthernet0/1
P 10.10.20.0/24, 1 successors, FD is 2297856
        via 172.16.10.2 (2297856/128256), Serial0/0
        via 172.16.10.6 (2300416/156160), Serial0/1
P 10.10.30.0/24, 1 successors, FD is 2297856
        via 172.16.10.2 (2297856/128256), Serial0/0
        via 172.16.10.6 (2300416/156160), Serial0/1
P 10.10.40.0/24, 1 successors, FD is 2297856
        via 172.16.10.6 (2297856/128256), Serial0/1
        via 172.16.10.2 (2300416/156160), Serial0/0
P 10.10.50.0/24, 1 successors, FD is 2297856
        via 172.16.10.6 (2297856/128256), Serial0/1
        via 172.16.10.2 (2300416/156160), Serial0/0
P 192.168.10.0/24, 2 successors, FD is 2172416
        via 172.16.10.2 (2172416/28160), Serial0/0
        via 172.16.10.6 (2172416/28160), Serial0/1
P 172.16.10.4/30, 1 successors, FD is 2169856
        via Connected, Serial0/1
P 172.16.10.0/30, 1 successors, FD is 2169856
        via Connected, Serial0/0
```

Notice that every route in this output is preceded by a *P*, which shows that these routes are in a *passive state*. This is good because routes in the active state indicate that the router has lost its path to this network and is searching for a replacement. Each entry also reveals the feasible distance, or FD, to each remote network, as well as, the next-hop neighbor through which packets will travel to this destination. Each entry also has two numbers in brackets, with the first indicating the feasible distance and the second, the advertised distance to a remote network.

Again, here's our equal- and unequal-cost load-balancing output shown in the topology table:

```
Corp#sh ip eigrp top | section 192.168.10.0
P 192.168.10.0/24, 2 successors, FD is 2172416
        via 172.16.10.2 (2172416/28160), Serial0/0
        via 172.16.10.6 (2172416/28160), Serial0/1
```

The preceding output shows equal-cost load balancing, and here is our unequal-cost load balancing in action:

```
Corp#sh ip eigrp top | section 10.10.50.0
P 10.10.50.0/24, 1 successors, FD is 2297856
        via 172.16.10.6 (2297856/128256), Serial0/1
        via 172.16.10.2 (2300416/156160), Serial0/0
```

The command `show ip eigrp traffic` enables us to see if updates are being sent. If the counters for EIGRP input and output packets don't increase, it means that no EIGRP information is being sent between peers. The following output indicates that the Corp router is experiencing normal traffic:

```
Corp#show ip eigrp traffic
IP-EIGRP Traffic Statistics for process 200
  Hellos sent/received: 2208/2310
  Updates sent/received: 184/183
  Queries sent/received: 17/4
  Replies sent/received: 4/18
  Acks sent/received: 62/65
  Input queue high water mark 2, 0 drops
```

All of the packet types I talked about in the section on RTP are represented in the output of this command. And we can't forget the always useful troubleshooting command `show ip protocols`. Here's the output the Corp router gives us after using it:

```
Routing Protocol is "eigrp 20"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Default networks flagged in outgoing updates
  Default networks accepted from incoming updates
  EIGRP metric weight K1=1, K2=0, K3=1, K4=0, K5=0
  EIGRP maximum hopcount 100
  EIGRP maximum metric variance 2
  Redistributing: eigrp 20
  EIGRP NSF-aware route hold timer is 240s
  Automatic network summarization is not in effect
  Maximum path: 4
  Routing for Networks:
    10.0.0.0
    172.16.0.0
  Routing Information Sources:
    Gateway         Distance      Last Update
    (this router)         90      04:23:51
    172.16.10.6           90      02:30:48
    172.16.10.2           90      02:30:48
  Distance: internal 90 external 170
```

In this output, we can see that EIGRP is enabled for autonomous system 20 and that the K values are set to their defaults. The variance is 2, so both equal- and unequal-cost load balancing is happening here. Automatic summarization has been turned off. We can also see that EIGRP is advertising two classful networks and that it sees two neighbors.

The `show ip eigrp events` command displays a log of every EIGRP event: when routes are injected and removed from the routing table and when EIGRP adjacencies are reset or fail. This information is so helpful in determining if there are routing instabilities in the network! Be advised that this command can result in quite a flood of information even for really simple configurations like ours. To demonstrate, here's the output the Corp router divulged after I used it:

```
Corp#show ip eigrp events
Event information for AS 20:
1    22:24:24.258 Metric set: 172.16.10.0/30 2169856
2    22:24:24.258 FC sat rdbmet/succmet: 2169856 0
3    22:24:24.258 FC sat nh/ndbmet: 0.0.0.0 2169856
4    22:24:24.258 Find FS: 172.16.10.0/30 2169856
5    22:24:24.258 Metric set: 172.16.10.4/30 2169856
6    22:24:24.258 FC sat rdbmet/succmet: 2169856 0
7    22:24:24.258 FC sat nh/ndbmet: 0.0.0.0 2169856
8    22:24:24.258 Find FS: 172.16.10.4/30 2169856
9    22:24:24.258 Metric set: 192.168.10.0/24 2172416
10   22:24:24.258 Route install: 192.168.10.0/24 172.16.10.2
11   22:24:24.258 Route install: 192.168.10.0/24 172.16.10.6
12   22:24:24.254 FC sat rdbmet/succmet: 2172416 28160
13   22:24:24.254 FC sat nh/ndbmet: 172.16.10.6 2172416
14   22:24:24.254 Find FS: 192.168.10.0/24 2172416
15   22:24:24.254 Metric set: 10.10.50.0/24 2297856
16   22:24:24.254 Route install: 10.10.50.0/24 172.16.10.6
17   22:24:24.254 FC sat rdbmet/succmet: 2297856 128256
18   22:24:24.254 FC sat nh/ndbmet: 172.16.10.6 2297856
19   22:24:24.254 Find FS: 10.10.50.0/24 2297856
20   22:24:24.254 Metric set: 10.10.40.0/24 2297856
21   22:24:24.254 Route install: 10.10.40.0/24 172.16.10.6
22   22:24:24.250 FC sat rdbmet/succmet: 2297856 128256
 --More--
```

Troubleshooting Example with EIGRP

Throughout this chapter I've covered many of the problems that commonly occur with EIGRP and how to verify and troubleshoot these issues. Make sure you clearly understand what I have shown you so far in this chapter so you're prepared to answer any question the Cisco exam could possibly throw at you!

Just to make sure you're solidly armed with all the skills you need to ace the exam, as well as, successfully administer a network, I'm going to provide even more examples about verifying EIGRP. We'll be dealing with mostly the same commands and problems we've already covered, but this is so important and the best way to get this all nailed down is to practice troubleshooting an EIGRP network as much as possible!

With that, after you've configured EIGRP, you would first test connectivity to the remote network by using the Ping program. If that fails, you need to check whether the directly connected router is in the neighbor table.

Here are some key things to look for if neighbors haven't formed an adjacency:

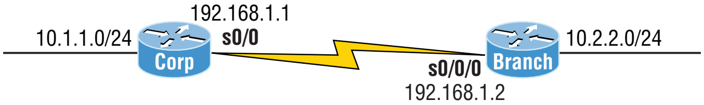
- Interfaces between the devices are down.
- The two routers have mismatching EIGRP autonomous system numbers.
- Proper interfaces are not enabled for the EIGRP process.
- An interface is configured as passive.
- The K values are mismatched.
- EIGRP authentication is misconfigured.

Also, if the adjacency is up, but you're not receiving remote network updates, there may be a routing problem, likely caused by these issues:

- The proper networks aren't being advertised under the EIGRP process.
- An access list is blocking the advertisements from remote networks.
- Automatic summary is causing confusion in your discontiguous network.

Let's use Figure 5.10 as our example network and run through some troubleshooting scenarios. I've preconfigured the routers with IP addresses, and without having to try too hard, I also snuck in a few snags for us to find and fix. Let's see what we're facing.

FIGURE 5.10 Troubleshooting scenario



A good place to start is by checking to see if we have an adjacency with `show ip eigrp neighbors` and `show ip eigrp interfaces`. It's also smart to see what information the `show ip eigrp topology` command reveals:

```
Corp#sh ip eigrp neighbors
IP-EIGRP neighbors for process 20
```

```
Corp#

Corp#sh ip eigrp interfaces
IP-EIGRP interfaces for process 20

                       Xmit Queue   Mean   Pacing Time   Multicast    Pending
Interface        Peers  Un/Reliable  SRTT   Un/Reliable   Flow Timer   Routes
Se0/1              0        0/0         0       0/15         50           0
Fa0/0              0        0/0         0       0/1           0           0
Se0/0              0        0/0         0       0/15         50           0

Corp#sh ip eigrp top
IP-EIGRP Topology Table for AS(20)/ID(10.10.11.1)

Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply,
       r - reply Status, s - sia Status

P 10.1.1.0/24, 1 successors, FD is 28160
        via Connected, FastEthernet0/0
```

Alright—we can see by looking at the neighbor and the interface as well as the topology table command that our LAN is up on the Corp router but the serial link isn't working between routers because we don't have an adjacency. From the `show ip eigrp interfaces` command, we can establish that EIGRP is running on all interfaces, so that means our network statements under the EIGRP process are probably correct, but we'll verify that later.

Let's move on by checking into our Physical and Data Link status with the `show ip int brief` command because maybe there's a physical problem between routers:

```
Corp#sh ip int brief
Interface                IP-Address    OK? Method Status                Protocol
FastEthernet0/0          10.1.1.1      YES manual up                       up
Serial0/0                192.168.1.1   YES manual up                       up
FastEthernet0/1          unassigned    YES manual administratively down down
Serial0/1                172.16.10.5   YES manual administratively down down
Corp#
Corp#sh protocols s0/0
Serial0/0 is up, line protocol is up
  Internet address is 192.168.1.1/30
Corp#
```

Well, since the Serial0/0 interface shows the correct IP address and the status is up/up, it means we have a good Data Link connection between routers, so it's not a physical link issue between the routers, which is good! Notice I also used the `show protocols` command, which gave me the subnet mask for the link. Remember, the information obtained via the two commands gives us only layer 1 and layer 2 status and doesn't mean we can ping across the link. In other words, we might have a layer 3 issue, so let's check the Branch router with the same commands:

```
Branch#sh ip int brief
Interface                  IP-Address    OK? Method Status                Protocol
FastEthernet0/0            10.2.2.2      YES manual up                       up
FastEthernet0/1            unassigned    YES manual administratively down down
Serial0/0/0                192.168.1.2   YES manual up                       up
Serial0/0/1                unassigned    YES unset  administratively down down
Branch#
Branch#sh proto s0/0/0
Serial0/0/0 is up, line protocol is up
  Internet address is 192.168.1.2/30
Branch#
```

Okay, well, we can see that our IP address and mask are correct, and that the link shows up/up, so we're looking pretty good! Let's try to ping from the Corp router to the Branch router now:

```
Corp#ping 192.168.1.2


Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/3/4 ms
Corp#
```

Now because that was successful, we've ruled out layer 1, 2, or 3 issues between routers at this point! Since everything seems to be working between the routers, except EIGRP, checking our EIGRP configurations is our next move. Let's start with the `show ip protocols` command:

```
Corp#sh ip protocols
Routing Protocol is "eigrp 20"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Default networks flagged in outgoing updates
  Default networks accepted from incoming updates
  EIGRP metric weight K1=1, K2=0, K3=1, K4=0, K5=0
  EIGRP maximum hopcount 100
  EIGRP maximum metric variance 2
  Redistributing: eigrp 20
  EIGRP NSF-aware route hold timer is 240s
  Automatic network summarization is in effect
  Maximum path: 4
  Routing for Networks:
    10.0.0.0
    172.16.0.0
    192.168.1.0
Passive Interface(s):
    FastEthernet0/1
  Routing Information Sources:
    Gateway         Distance      Last Update
    (this router)         90      20:51:48
    192.168.1.2           90      00:22:58
    172.16.10.6           90      01:58:46
    172.16.10.2           90      01:59:52
  Distance: internal 90 external 170
```

This output shows us we're using AS 20, that we don't have an access-list filter list set on the routing tables, and that our K values are set to default. We can see that we're routing for the 10.0.0.0, 172.16.0.0, and 192.168.1.0 networks and that we have a passive interface on interface FastEthernet0/1. We don't have an interface configured for the 172.16.0.0 network, which means that this entry is an extra network statement under EIGRP. But that won't hurt anything, so this is not causing our issue. Last, the passive interface is not causing a problem with this network either, because we're not using interface Fa0/1. Still, keep in mind that when troubleshooting, it's always good to see if there are any interfaces set to passive.

Let's see what the `show interface` command will tell us:

```
Corp#sh interfaces s0/0
Serial0/0 is up, line protocol is up
  Hardware is PowerQUICC Serial
  Description: <<Connection to Branch>>
  Internet address is 192.168.1.1/30
  MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation HDLC, loopback not set
[output cut]
```

Looks like our statistics are set to defaults, so nothing really pops as a problem here. But remember when I covered the steps to check if there is no adjacency back at the beginning of this section? In case you forgot, here's a list of things to investigate:

- The interface between the devices are down.
- The two routers have mismatching EIGRP autonomous system numbers.
- The proper interfaces aren't enabled for the EIGRP process.
- An interface is configured as passive.
- K values are mismatched.
- EIGRP authentication is misconfigured.

Okay, our interfaces are not down, our AS number matches, layer 3 is working between routers, all the interfaces show up under the EIGRP process, and none of our needed interfaces are passive, so now we'll have to look even deeper into the EIGRP configuration to uncover the problem.

Since the Corp router has the basic default configurations, we need to check the Branch router's EIGRP configuration:

```
Branch#sh ip protocols
Routing Protocol is "eigrp 20"
  Outgoing update filter list for all interfaces is 10
  Incoming update filter list for all interfaces is not set
  Default networks flagged in outgoing updates
  Default networks accepted from incoming updates
  EIGRP metric weight K1=1, K2=0, K3=0, K4=0, K5=0
  EIGRP maximum hopcount 100
  EIGRP maximum metric variance 1
  Redistributing: eigrp 20
  EIGRP NSF-aware route hold timer is 240s
  Automatic network summarization is not in effect
  Maximum path: 4
  Routing for Networks:
    10.0.0.0
    192.168.1.0
  Routing Information Sources:
    Gateway         Distance      Last Update
    192.168.1.1           90      00:27:09
  Distance: internal 90 external 170
```

This router has the correct AS—always check this first—and we're routing for the correct networks. But I see two possible snags here, do you? First, the outgoing ACL filter list is set, but the metrics are not set to default. Remember, just because an ACL is set doesn't mean it's automatically giving you grief. Second, the K values must match, and we know these values are not matching the Corp router!

Let's take a look at the Branch interface statistics to see what else might be wrong:

```
Branch>sh int s0/0/0
Serial0/0/0 is up, line protocol is up
  Hardware is GT96K Serial
  Internet address is 192.168.1.2/30
  MTU 1500 bytes, BW 512 Kbit, DLY 30000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation HDLC, loopback not set
  [output cut]
```

Aha! The bandwidth and delay are not set to their defaults and don't match the directly connected Corp router. Let's start by changing those back to the default and see if that fixes our problem:

```
Branch#config t
Branch(config)#int s0/0/0
Branch(config-if)#no bandwidth
Branch(config-if)#no delay
```

And let's check out our stats now to see if we're back to defaults:

```
Branch#sh int s0/0/0
Serial0/0/0 is up, line protocol is up
  Hardware is GT96K Serial
  Internet address is 192.168.1.2/30
  MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation HDLC, loopback not set
[output cut]
```

The bandwidth and delay are now at the defaults, so let's check our adjacencies next:

```
Corp#sh ip eigrp neighbors
IP-EIGRP neighbors for process 20
Corp#
```

Okay, so it wasn't the bandwidth and delay settings because our adjacency didn't come up, so let's set our K values back to default like this:

```
Branch#config t
Branch(config)#router eigrp 20
Branch(config-router)#metric weights 0 1 0 1 0 0
Branch(config-router)#do sho ip proto
Routing Protocol is "eigrp 20"
```

```
  Outgoing update filter list for all interfaces is 10
  Incoming update filter list for all interfaces is not set
  Default networks flagged in outgoing updates
  Default networks accepted from incoming updates
  EIGRP metric weight K1=1, K2=0, K3=1, K4=0, K5=0
[output cut]
```

I know this probably seems a little complicated at first, but it's something you shouldn't have to do much, if ever. Remember, there are five K values, so why 6 numbers? The first number listed is type of service (ToS), so always just set that to 0, which means you must type in six numbers as shown in my configuration example. After we chose the default of 0 first, the default K values are then 1 0 1 0 0, which is bandwidth and delay enabled. Let's check our adjacency now:

```
Corp#sh ip eigrp neighbors
IP-EIGRP neighbors for process 20
H   Address                 Interface       Hold Uptime   SRTT   RTO  Q  Seq
                                            (sec)         (ms)       Cnt Num
0   192.168.1.2             Se0/0             14 00:02:09    7   200  0  18
```

Bam! There we go! Looks like mismatched K values were our problem. Now let's just check to make sure we can ping from end to end and we're done:

```
Corp#ping 10.2.2.2


Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.2.2.2, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
Corp#
```

Rats! It looks like even though we have our adjacency, we still can't reach our remote network. Next step? Let's see what the routing table shows us:

```
Corp#sh ip route
[output cut]


     10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C       10.1.1.0/24 is directly connected, FastEthernet0/0
D       10.0.0.0/8 is a summary, 00:18:55, Null0
     192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
C       192.168.1.0/30 is directly connected, Serial0/0
D       192.168.1.0/24 is a summary, 00:18:55, Null0
```

The problem is screamingly clear now because I went through this in detail throughout this chapter. But just in case you still can't find it, let's look at the `show ip protocols` command output:

```
Routing Protocol is "eigrp 20"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Default networks flagged in outgoing updates
  Default networks accepted from incoming updates
  EIGRP metric weight K1=1, K2=0, K3=1, K4=0, K5=0
  EIGRP maximum hopcount 100
  EIGRP maximum metric variance 2
  Redistributing: eigrp 20
  EIGRP NSF-aware route hold timer is 240s
  Automatic network summarization is in effect
  Automatic address summarization:
    192.168.1.0/24 for FastEthernet0/0
      Summarizing with metric 2169856
    10.0.0.0/8 for Serial0/0
      Summarizing with metric 28160
 [output cut]
```

By looking at the Figure 5.10 you should have noticed right away that we had a discontiguous network. This means that unless they are running 15.0 IOS code, the routers will auto-summarize, so we need to disable auto-summary:

```
Branch(config)#router eigrp 20
Branch(config-router)#no auto-summary
008412:%DUAL-5-NBRCHANGE:IP-EIGRP(0) 20:Neighbor 192.168.1.1 (Serial0/0/0) is
resync:
peer graceful-restart
```

```
Corp(config)#router eigrp 20
Corp(config-router)#no auto-summary
Corp(config-router)#
*Feb 27 19:52:54:%DUAL-5-NBRCHANGE: IP-EIGRP(0) 20:Neighbor 192.168.1.2
(Serial0/0)
 is resync: summary configured
*Feb 27 19:52:54.177:IP-EIGRP(Default-IP-Routing-Table:20):10.1.1.0/24 - do
advertise
 out Serial0/0
*Feb 27 19:52:54:IP-EIGRP(Default-IP-Routing-Table:20):Int 10.1.1.0/24 metric
2816
```

```
0 - 25600 2560
*Feb 27 19:52:54:IP-EIGRP(Default-IP-Routing-Table:20):192.168.1.0/30 - do
advertise out Serial0/0
*Feb 27 19:52:54:IP-EIGRP(Default-IP-Routing-Table:20):192.168.1.0/24 - do
advertise out Serial0/0
*Feb 27 19:52:54:IP-EIGRP(Default-IP-Routing-Table:20):Int 192.168.1.0/24 metric
4294967295 - 0 4294967295
*Feb 27 19:52:54:IP-EIGRP(Default-IP-Routing-Table:20):10.0.0.0/8 - do advertise
 out Serial0/0
Corp(config-router)#
*Feb 27 19:52:54:IP-EIGRP(Default-IP-Routing-Table:20):Int 10.0.0.0/8 metric
4294967295 - 0 4294967295
*Feb 27 19:52:54:IP-EIGRP(Default-IP-Routing-Table:20):Processing incoming REPLY
packet
*Feb 27 19:52:54:IP-EIGRP(Default-IP-Routing-Table:20):Int 192.168.1.0/24 M
4294967295 - 1657856 4294967295 SM 4294967295 - 1657856 4294967295
*Feb 27 19:52:54:IP-EIGRP(Default-IP-Routing-Table:20):Int 10.0.0.0/8 M
4294967295 - 25600 4294967295 SM 4294967295 - 25600 4294967295
*Feb 27 19:52:54:IP-EIGRP(Default-IP-Routing-Table:20):Processing incoming UPDATE
packet
```

Finally the Corp looks happy, so it looks like we're good to go! Let's just check our routing table to be sure:

```
Corp#sh ip route
[output cut]
     10.0.0.0/24 is subnetted, 1 subnets
C       10.1.1.0 is directly connected, FastEthernet0/0
     192.168.1.0/30 is subnetted, 1 subnets
C       192.168.1.0 is directly connected, Serial0/0
```

What the heck? How can this be! We saw all those updates on the Corp console, right? Let's check the configuration of EIGRP by looking at the active configuration on the Branch router:

```
Branch#sh run
[output cut]
!
router eigrp 20
 network 10.0.0.0
 network 192.168.1.0
 distribute-list 10 out
 no auto-summary
!
```

We can see that the access list is set outbound on the routing table of the Branch router. This may be preventing us from receiving the updates from remote networks! Let's see what the ACL 10 list is doing:

```
Branch#sh access-lists
Standard IP access list 10
    10 deny   any (40 matches)
    20 permit any
```

Now who in the world would stick an access list like this on a router? This ACL says to deny every packet, which makes the second line of the ACL irrelevant since every single packet will match the first line! This has got to be the source of our troubles, so let's remove that list and see if the Corp router starts working:

```
Branch#config t
Branch(config)#router eigrp 20
Branch(config-router)#no distribute-list 10 out
```

Okay, with that ugly thing gone, let's check to see if we're receiving our remote networks now:

```
Corp#sh ip route
[output cut]
     10.0.0.0/24 is subnetted, 2 subnets
D       10.2.2.0 [90/2172416] via 192.168.1.2, 00:00:24, Serial0/0
C       10.1.1.0 is directly connected, FastEthernet0/0
     192.168.1.0/30 is subnetted, 1 subnets
C       192.168.1.0 is directly connected, Serial0/0
Corp#
Corp#ping 10.2.2.2


Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.2.2.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/3/4 ms
Corp#
```

Clear skies! We're up and running. We had mismatched K values, discontiguous networking, and a nasty ACL on our routing table. For the CCNA R/S objectives, always check for an ACL on the actual interface as well, not just in the routing table. It could be set on the interface or routing table, either one, or both! And never forget to check for passive interfaces when troubleshooting a routing protocol issue!

All of these commands are seriously powerful tools in the hands of a savvy professional faced with the task of troubleshooting myriad network issues. I could go on and on about the profusion of information these commands can generate and how well they can equip us to solve virtually every networking ill, but that would be way outside the scope of this book. Even so, I have no doubt that the foundation I've given you here will prove practical and valuable for certification purposes, as well as, when working in the real networking world.

Now it's time to relax a bit as we move into the easiest part of this chapter, seriously—not joking! You still need to pay attention though.

EIGRPv6

Visit `ccna.gg/ch5/b` *for a companion MicroNugget from CBT Nuggets.*

As I just was saying, welcome to the easiest part of the chapter! Of course, I only mostly mean that, and here's why: I talked about IPv6 in the earlier ICND1 chapters, and in order to continue on with this section of the chapter, you need to have that vital, foundational part of IPv6 down solidly before you dare to dwell here! If you do, you're pretty much set and this will all be pretty simple for you.

EIGRPv6 works much the same way as its IPv4 predecessor does—most of the features that EIGRP provided before EIGRPv6 will still be available.

EIGRPv6 is still an advanced distance-vector protocol that has some link-state features. The neighbor discovery process using Hellos still happens, and it still provides reliable communication with Reliable Transport Protocol that gives us loop-free fast convergence using the Diffusing Update Algorithm (DUAL).

Hello packets and updates are sent using multicast transmission, and as with RIPng, EIGRPv6's multicast address stayed almost the same. In IPv4 it was 224.0.0.10; in IPv6, it's FF02::A (A = 10 in hexadecimal notation).

But clearly, there are key differences between the two versions. Most notably the use of the pesky `network` command is gone, so it's hard to make a mistake with EIGRPv6. Also, the network and interface to be advertised must be enabled from interface configuration mode with one simple command.

But you still have to use the router configuration mode to enable the routing protocol in EIGRPv6 because the routing process must be literally enabled like an interface with the `no shutdown` command—interesting! However, the 15.0 code does enable this by default, so this command actually may or may not be needed.

Here's an example of enabling EIGRPv6 on the Corp router:

```
Corp(config)#ipv6 unicast-routing
Corp(config)#ipv6 router eigrp 10
```

The 10 in this case is still the AS number. The prompt changes to `(config-rtr)`, and from here, just initiate a `no shutdown` if needed:

```
Corp(config-rtr)#no shutdown
```

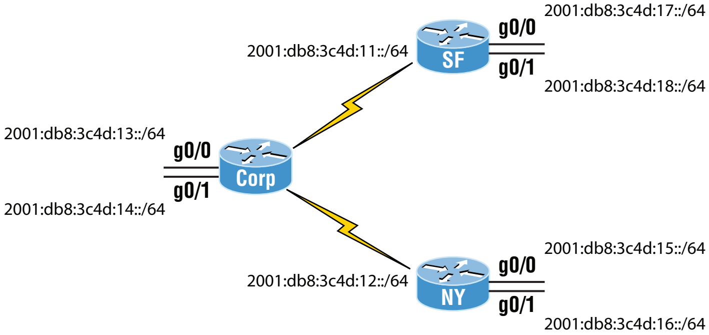
Other options also can be configured in this mode, like redistribution and router ID (RID). So now, let's go to the interface and enable IPv6:

```
Corp(config-if)#ipv6 eigrp 10
```

The 10 in the interface command again references the AS number that was enabled in the configuration mode.

Figure 5.11 shows the layout we've been using throughout this chapter, only with IPv6 addresses now assigned to interfaces. I used the EUI-64 option on each interface so each router assigned themselves an IPv6 address after I typed in the 64-bit network/subnet address.

FIGURE 5.11 Configuring EIGRPv6 on our Internetwork


2001:db8:3c4d:17::/64
g0/0
2001:db8:3c4d:11::/64
SF
g0/1
2001:db8:3c4d:18::/64
2001:db8:3c4d:13::/64
g0/0
Corp
g0/1
2001:db8:3c4d:14::/64
2001:db8:3c4d:15::/64
g0/0
2001:db8:3c4d:12::/64
NY
g0/1
2001:db8:3c4d:16::/64

We'll start with the Corp router. Really, all we need to know in order to enable EIGRPv6 are which interfaces we're using and want to advertise our networks.

```
Corp#config t
Corp(config)#ipv6 router eigrp 10
Corp(config-rtr)#no shut
Corp(config-rtr)#router-id 1.1.1.1
Corp(config-rtr)#int s0/0/0
Corp(config-if)#ipv6 eigrp 10
Corp(config-if)#int s0/0/1
Corp(config-if)#ipv6 eigrp 10
Corp(config-if)#int g0/0
Corp(config-if)#ipv6 eigrp 10
Corp(config-if)#int g0/1
Corp(config-if)#ipv6 eigrp 10
```

I had erased and reloaded the routers before I started this EIGRPv6 section of the chapter. What this means is that there were no 32-bit addresses on the router in order to create the RID for EIGRP, so I had to set it under the IPv6 router global command, which is the same command used with EIGRP and EIGRPv6. Unlike OSPF, the RID isn't that important, and it can actually be the same address on every router. You just can't get away with doing this with OSPF! The configuration for EIGRPv6 was pretty straightforward because unless you type the AS number wrong, it's pretty hard to screw this up!

Okay, let's configure the SF and NY routers now, and then we'll verify our networks:

```
SF#config t
SF(config)#ipv6 router eigrp 10
SF(config-rtr)#no shut
SF(config-rtr)#router-id 2.2.2.2
SF(config-rtr)#int s0/0/0
SF(config-if)#ipv6 eigrp 10
SF(config-if)#int g0/0
SF(config-if)#ipv6 eigrp 10
SF(config-if)#int g0/1
SF(config-if)#ipv6 eigrp 10


NY#config t
NY(config)#ipv6 router eigrp 10
NY(config-rtr)#no shut
NY(config-rtr)#router-id 3.3.3.3
NY(config-rtr)#int s0/0/0
NY(config-if)#ipv6 eigrp 10
NY(config-if)#int g0/0
NY(config-if)#ipv6 eigrp 10
NY(config-if)#int g0/1
```

Since we configured EIGRPv6 on a per-interface basis, no worries about having to use the `passive-interface` command. This is because if we don't enable the routing protocol on an interface, it's just not part of the EIGRPv6 process. We can see which interfaces are part of the EIGRPv6 process with the `show ipv6 eigrp interfaces` command like this:

```
Corp#sh ipv6 eigrp interfaces
IPv6-EIGRP interfaces for process 10
                        Xmit Queue   Mean   Pacing Time   Multicast    Pending
Interface        Peers  Un/Reliable  SRTT   Un/Reliable   Flow Timer   Routes
Se0/0/0            1        0/0      1236       0/10           0           0
Se0/0/1            1        0/0      1236       0/10           0           0
Gig0/1             0        0/0      1236       0/10           0           0
Gig0/0             0        0/0      1236       0/10           0           0
Corp#
```

Looks great so far—all the interfaces we want in our AS are listed, so we're looking good for our Corp's local configuration. Now it's time to check if our adjacencies came up with the `show ipv6 eigrp neighbors` command:

```
Corp#sh ipv6 eigrp neighbors
IPv6-EIGRP neighbors for process 10
H   Address                 Interface     Hold   Uptime   SRTT   RTO  Q  Seq
                                          (sec)           (ms)       Cnt Num
0   Link-local address:       Se0/0/0      10   00:01:40  40     1000  0   11
    FE80::201:C9FF:FED0:3301
1   Link-local address:       Se0/0/1      14   00:01:24  40     1000  0   11
    FE80::209:7CFF:FE51:B401
```

It's great that we can see neighbors listed off of each serial interface, but do you notice something missing from the preceding output? That's right, the actual IPv6 network/subnet addresses of the links aren't listed in the neighbor table! Only the link-local addresses are used for forming EIGRP neighbor adjacencies. With IPv6, neighbor interfaces and next-hop addresses are always link-local.

We can verify our configuration with the `show ip protocols` command:

```
Corp#sh ipv6 protocols
IPv6 Routing Protocol is "connected"
IPv6 Routing Protocol is "static
IPv6 Routing Protocol is "eigrp  10 "
  EIGRP metric weight K1=1, K2=0, K3=1, K4=0, K5=0
  EIGRP maximum hopcount 100
  EIGRP maximum metric variance 1
  Interfaces:
    Serial0/0/0
    Serial0/0/1
    GigabitEthernet0/0
    GigabitEthernet0/1
Redistributing: eigrp 10
  Maximum path: 16
  Distance: internal 90 external 170
```

You can verify the AS number from this output, but be sure to verify your K values, variance, and interfaces, too. Remember that the AS number and interfaces are the first factors to check when troubleshooting.

The topology table lists all feasible routes in the network, so this output can be rather long, but let's see what this shows us:

```
Corp#sh ipv6 eigrp topology
IPv6-EIGRP Topology Table for AS 10/ID(1.1.1.1)
```

```
Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply,
       r - Reply status


P 2001:DB8:C34D:11::/64, 1 successors, FD is 2169856
         via Connected, Serial0/0/0
P 2001:DB8:C34D:12::/64, 1 successors, FD is 2169856
         via Connected, Serial0/0/1
P 2001:DB8:C34D:14::/64, 1 successors, FD is 2816
         via Connected, GigabitEthernet0/1
P 2001:DB8:C34D:13::/64, 1 successors, FD is 2816
         via Connected, GigabitEthernet0/0
P 2001:DB8:C34D:17::/64, 1 successors, FD is 2170112
         via FE80::201:C9FF:FED0:3301 (2170112/2816), Serial0/0/0
P 2001:DB8:C34D:18::/64, 1 successors, FD is 2170112
         via FE80::201:C9FF:FED0:3301 (2170112/2816), Serial0/0/0
P 2001:DB8:C34D:15::/64, 1 successors, FD is 2170112
         via FE80::209:7CFF:FE51:B401 (2170112/2816), Serial0/0/1
P 2001:DB8:C34D:16::/64, 1 successors, FD is 2170112
         via FE80::209:7CFF:FE51:B401 (2170112/2816), Serial0/0/1
```

Since we only have eight networks in our internetwork, we can see all eight networks in the topology table, which clearly is as it should be. I've highlighted a couple of things I want to discuss, and the first is that you need to be able to read and understand a topology table. This includes understanding which routes are directly connected and which are being advertised via neighbors. The `via Connected` shows us our directly connected networks. The second item I want to show you is `(2170112/2816)`, which is the FD/AD, and by the way, it's no different than if you're working with IPv4.

So let's wrap up this chapter by taking a look at a routing table:

```
Corp#sh ipv6 route eigrp
IPv6 Routing Table - 13 entries
Codes: C - Connected, L - Local, S - Static, R - RIP, B - BGP
       U - Per-user Static route, M - MIPv6
       I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary
       O - OSPF intra, OI - OSPF inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2
       ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2
       D - EIGRP, EX - EIGRP external
C   2001:DB8:C34D:11::/64 [0/0]
     via ::, Serial0/0/0
L   2001:DB8:C34D:11:230:A3FF:FE36:B101/128 [0/0]
     via ::, Serial0/0/0
```

```
C   2001:DB8:C34D:12::/64 [0/0]
     via ::, Serial0/0/1
L   2001:DB8:C34D:12:230:A3FF:FE36:B102/128 [0/0]
     via ::, Serial0/0/1
C   2001:DB8:C34D:13::/64 [0/0]
     via ::, GigabitEthernet0/0
L   2001:DB8:C34D:13:2E0:F7FF:FEDA:7501/128 [0/0]
     via ::, GigabitEthernet0/0
C   2001:DB8:C34D:14::/64 [0/0]
     via ::, GigabitEthernet0/1
L   2001:DB8:C34D:14:2E0:F7FF:FEDA:7502/128 [0/0]
     via ::, GigabitEthernet0/1
D   2001:DB8:C34D:15::/64 [90/2170112]
     via FE80::209:7CFF:FE51:B401, Serial0/0/1
D   2001:DB8:C34D:16::/64 [90/2170112]
     via FE80::209:7CFF:FE51:B401, Serial0/0/1
D   2001:DB8:C34D:17::/64 [90/2170112]
     via FE80::201:C9FF:FED0:3301, Serial0/0/0
D   2001:DB8:C34D:18::/64 [90/2170112]
     via FE80::201:C9FF:FED0:3301, Serial0/0/0
L   FF00::/8 [0/0]
     via ::, Null0
```

I highlighted the EIGRPv6 injected routes that were injected into the routing table. It's important to notice that in order for IPv6 to get to a remote network, the router uses the next hop link-local address. Do you see that in the table? For example, `via FE80::209:7CFF:FE51:B401, Serial0/0/1` is the link-local address of the NY router.

See? I told you it was easy!

Summary

It's true that this chapter has been pretty extensive, so let's briefly recap what we covered in it. EIGRP, the main focus of the chapter, is a hybrid of link-state routing and typically referred to as an advanced distance-vector protocol. It allows for unequal-cost load balancing, controlled routing updates, and formal neighbor adjacencies called relationships to be formed.

EIGRP uses the capabilities of the Reliable Transport Protocol (RTP) to communicate between neighbors and utilizes the Diffusing Update Algorithm (DUAL) to compute the best path to each remote network.

We also covered the configuration of EIGRP and explored a number of troubleshooting commands plus key ways and means to help solve some common networking issues.

Moving on, EIGRP facilitates unequal-cost load balancing, controlled routing updates, and formal neighbor adjacencies.

I also went over the configuration of EIGRP and explored a number of troubleshooting commands, as well as, taking you through a highly informative scenario that will not only help you to ace the exam, it will help you confront and overcome many troubleshooting issues common to today's internetworks!

Finally, I went over the easiest section at the end of this long chapter: EIGRPv6. Easy to understand, configure, and verify!

Exam Essentials

Know EIGRP features. EIGRP is a classless, advanced distance-vector protocol that supports IP and now IPv6. EIGRP uses a unique algorithm, called DUAL, to maintain route information and uses RTP to communicate with other EIGRP routers reliably.

Know how to configure EIGRP. Be able to configure basic EIGRP. This is configured the same as RIP with classful addresses.

Know how to verify EIGRP operation. Know all of the EIGRP `show` commands and be familiar with their output and the interpretation of the main components of their output.

Be able to read an EIGRP topology table. Understand which are successors, which are feasible successors, and which routes will become successors if the main successor fails.

You must be able to troubleshoot EIGRP. Go through the EIGRP troubleshooting scenario and make sure you understand to look for the AS number, ACLs, passive interfaces, variance, and other factors.

Be able to read an EIGRP neighbor table. Understand the output of the `show ip eigrp neighbor` command.

Understand how to configure EIGRPv6. To configure EIGRPv6, first create the autonomous system from global configuration mode and perform a `no shutdown`. Then enable EIGRPv6 on each interface individually.

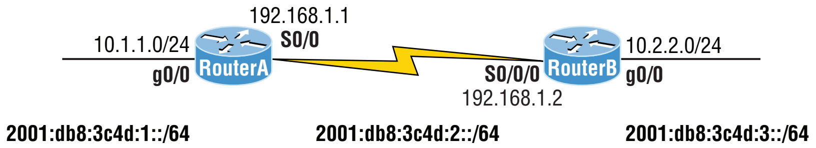
Written Lab 5

The answers to this lab can be found in Appendix A, "Answers to Written Labs."

1. What is the command to enable EIGRPv6 from global configuration mode?
2. What is the EIGRPv6 multicast address?
3. True/False: Each router within an EIGRP domain must use different AS numbers.
4. If you have two routers with various K-values assigned, what will this do to the link?
5. What type of EIGRP interface will neither send nor receive Hello packets?

Hands-on Labs

In this section, you will use the following network and add EIGRP and EIGRPv6 routing.



The first lab requires you to configure two routers for EIGRP and then view the configuration. In the last lab, you will be asked to enable EIGRPv6 routing on the same network. Note that the labs in this chapter were written to be used with real equipment—real cheap equipment that is. I wrote these labs with the cheapest, oldest routers I had lying around so you can see that you don't need expensive gear to get through some of the hardest labs in this book. However, you can use the free LammleSim IOS version simulator or Cisco's Packet Tracer to run through these labs.

The labs in this chapter are as follows:

Lab 5.1: Configuring and Verifying EIGRP

Lab 5.2: Configuring and Verifying EIGRPv6

Hands-on Lab 5.1: Configuring and Verifying EIGRP

This lab will assume you have configured the IP addresses on the interfaces as shown in the preceding diagram.

1. Implement EIGRP on RouterA.

```
RouterA#conf t
Enter configuration commands, one per line.
  End with CNTL/Z.
```

```
RouterA(config)#router eigrp 100
RouterA(config-router)#network 192.168.1.0
RouterA(config-router)#network 10.0.0.0
RouterA(config-router)#^Z
RouterA#
```

2. Implement EIGRP on RouterB.

```
RouterB#conf t
Enter configuration commands, one per line.
  End with CNTL/Z.
RouterB(config)#router eigrp 100
RouterB(config-router)#network 192.168.1.0
RouterA(config-router)#network 10.0.0.0
RouterB(config-router)#exit
RouterB#
```

3. Display the topology table RouterA.

```
RouterA#show ip eigrp topology
```

4. Display the routing table for RouterA.

```
RouterA #show ip route
```

5. Display the neighbor table for RouterA.

```
RouterA show ip eigrp neighbor
```

6. Type the command on each router to fix the routing problem. You did see a problem didn't you? Yes, the network is discontiguous.

```
RouterA#config t
RouterA(config)#router eigrp 100
RouterA(config-router)#no auto-summary


RouterB#config t
RouterA(config)#router eigrp 100
RouterA(config-router)#no auto-summary
```

7. Verify your routes with the `show ip route` command.

Hands-on Lab 5.2: Configuring and Verifying EIGRPv6

This lab will assume you configured the IPv6 address as shown in the diagram preceding Lab 5.1.

1. Implement EIGRPv6 on RouterA with AS 100.

```
RouterA#config t
RouterA (config)#ipv6 router eigrp 100
RouterA (config-rtr)#no shut
RouterA (config-rtr)#router-id 2.2.2.2
RouterA (config-rtr)#int s0/0
RouterA (config-if)#ipv6 eigrp 100
RouterA (config-if)#int g0/0
RouterA (config-if)#ipv6 eigrp 100
```

2. Implement EIGRP on RouterB.

```
RouterA#config t
RouterB(config)#ipv6 router eigrp 100
RouterB(config-rtr)#no shut
RouterB(config-rtr)#router-id 2.2.2.2
RouterB(config-rtr)#int s0/0
RouterB(config-if)#ipv6 eigrp 100
RouterB(config-if)#int g0/0
RouterB(config-if)#ipv6 eigrp 100
```

3. Display the topology table RouterA.

```
RouterA#show ipv6 eigrp topology
```

4. Display the routing table for RouterA.

```
RouterA #show ipv6 route
```

5. Display the neighbor table for RouterA.

```
RouterA show ipv6 eigrp neighbor
```

Review Questions

NOTE

The following questions are designed to test your understanding of this chapter's material. For more information on how to get additional questions, please see this book's introduction.

The answers to these questions can be found in Appendix B, "Answers to Chapter Review Questions."

1. There are three possible routes for a router to reach a destination network. The first route is from OSPF with a metric of 782. The second route is from RIPv2 with a metric of 4. The third is from EIGRP with a composite metric of 20514560. Which route will be installed by the router in its routing table?
   - **A.** RIPv2
   - **B.** EIGRP
   - **C.** OSPF
   - **D.** All three
2. Which EIGRP information is held in RAM and maintained through the use of Hello and update packets? (Choose two.)
   - **A.** Neighbor table
   - **B.** STP table
   - **C.** Topology table
   - **D.** DUAL table
3. What will be the reported distance to a downstream neighbor router for the 10.10.30.0 network, with the neighbor adding the cost to this neighbor to find the true FD?

```
P 10.10.30.0/24, 1 successors, FD is 2297856
        via 172.16.10.2 (2297856/128256), Serial0/0
```

   - **A.** Four hops
   - **B.** 2297856
   - **C.** 128256
   - **D.** EIGRP doesn't use reported distances.

4. Where are EIGRP successor routes stored?
   - **A.** In the routing table only
   - **B.** In the neighbor table only
   - **C.** In the topology table only
   - **D.** In the routing table and the neighbor table
   - **E.** In the routing table and the topology table
   - **F.** In the topology table and the neighbor table

5. Which command will display all the EIGRP feasible successor routes known to a router?
   - **A.** `show ip routes *`
   - **B.** `show ip eigrp summary`
   - **C.** `show ip eigrp topology`
   - **D.** `show ip eigrp adjacencies`
   - **E.** `show ip eigrp neighbors detail`

6. Which of the following commands are used when routing with EIGRP or EIGRPv6? (Choose three.)
   - **A.** `network 10.0.0.0`
   - **B.** `eigrp router-id`
   - **C.** `variance`
   - **D.** `router eigrp`
   - **E.** `maximum-paths`

7. Serial0/0 goes down. How will EIGRP send packets to the 10.1.1.0 network?

```
Corp#show ip eigrp topology
[output cut]
P 10.1.1.0/24, 2 successors, FD is 2681842
         via 10.1.2.2 (2681842/2169856), Serial0/0
         via 10.1.3.1 (2973467/2579243), Serial0/2
         via 10.1.3.3 (2681842/2169856), Serial0/1
```

   - **A.** EIGRP will put the 10.1.1.0 network into active mode.
   - **B.** EIGRP will drop all packets destined for 10.1.1.0.
   - **C.** EIGRP will just keep sending packets out s0/1.
   - **E.** EIGRP will use s0/2 as the successor and keep routing to 10.1.1.0.

8. What command do you use to enable EIGRPv6 on an interface?
   - **A.** `router eigrp as`
   - **B.** `ip router eigrp as`
   - **C.** `router eigrpv6 as`
   - **D.** `ipv6 eigrp as`

9. What command was typed in to have these two paths to network 10.10.50.0 in the routing table?

```
D       10.10.50.0 [90/2297856] via 172.16.10.6, 00:00:20, Serial0/1
                   [90/6893568] via 172.16.10.2, 00:00:20, Serial0/0
```

   - **A.** `maximum-paths 2`
   - **B.** `variance 2`
   - **C.** `variance 3`
   - **D.** `maximum-hops 2`

10. A route to network 10.10.10.0 goes down. How does EIGRP respond in the local routing table? (Choose two.)
    - **A.** It sends a poison reverse with a maximum hop of 16.
    - **B.** If there is a feasible successor, that is copied and placed into the routing table.
    - **C.** If a feasible successor is not found, a query will be sent to all neighbors asking for a path to network 10.10.10.0.
    - **D.** EIGRP will broadcast out all interfaces that the link to network 10.10.10.0 is down and that it is looking for a feasible successor.

11. You need the IP address of the devices with which the router has established an adjacency. Also, the retransmit interval and the queue counts for the adjacent routers need to be checked. What command will display the required information?
    - **A.** `show ip eigrp adjacency`
    - **B.** `show ip eigrp topology`
    - **C.** `show ip eigrp interfaces`
    - **D.** `show ip eigrp neighbors`

12. For some reason, you cannot establish an adjacency relationship on a common Ethernet link between two routers. Looking at the output shown here, what are the causes of the problem? (Choose two.)

```
RouterA##show ip protocols
Routing Protocol is "eigrp 20"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Default networks flagged in outgoing updates
  Default networks accepted from incoming updates
  EIGRP metric weight K1=1, K2=0, K3=1, K4=0, K5=0

RouterB##show ip protocols
Routing Protocol is "eigrp 220"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Default networks flagged in outgoing updates
  Default networks accepted from incoming updates
  EIGRP metric weight K1=1, K2=1, K3=1, K4=0, K5=0
```

A. EIGRP is running on RouterA and OSPF is running RouterB.

B. There is an ACL set on the routing protocol.

C. The AS numbers don't match.

D. There is no default network accepted from incoming updates.

E. The K-values don't match.

F. There is a passive interface set.

13. Which are true regarding EIGRP successor routes? (Choose two.)

A. A successor route is used by EIGRP to forward traffic to a destination.

B. Successor routes are saved in the topology table to be used if the primary route fails.

C. Successor routes are flagged as "active" in the routing table.

D. A successor route may be backed up by a feasible successor route.

E. Successor routes are stored in the neighbor table following the discovery process.

14. The remote RouterB router has a directly connected network of 10.255.255.64/27. Which two of the following EIGRP network statements could you use so this directly connected network will be advertised under the EIGRP process? (Choose two.)

A. `network 10.255.255.64`

B. `network 10.255.255.64 0.0.0.31`

C. `network 10.255.255.64 0.0.0.0`

D. `network 10.255.255.64 0.0.0.15`

15. RouterA and RouterB are connected via their Serial 0/0 interfaces, but they have not formed an adjacency. Based on the following output, what could be the problem?

```
RouterA#sh ip protocols
Routing Protocol is "eigrp 220"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Default networks flagged in outgoing updates
  Default networks accepted from incoming updates
  EIGRP metric weight K1=1, K2=0, K3=1, K4=0, K5=0
  EIGRP maximum hopcount 100
  EIGRP maximum metric variance 2
  Redistributing: eigrp 220
  EIGRP NSF-aware route hold timer is 240s
  Automatic network summarization is in effect
  Maximum path: 4
  Routing for Networks:
    10.0.0.0
    172.16.0.0
    192.168.1.0
  Routing Information Sources:
    Gateway         Distance      Last Update
    (this router)         90      20:51:48
    192.168.1.2           90      00:22:58
    172.16.10.6           90      01:58:46
    172.16.10.2           90      01:59:52
  Distance: internal 90 external 170

RouterB#sh ip protocols
Routing Protocol is "eigrp 220"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Default networks flagged in outgoing updates
  Default networks accepted from incoming updates
  EIGRP metric weight K1=1, K2=0, K3=1, K4=0, K5=0
  EIGRP maximum hopcount 100
  EIGRP maximum metric variance 2
  Redistributing: eigrp 220
  EIGRP NSF-aware route hold timer is 240s
  Automatic network summarization is in effect
  Maximum path: 4
  Routing for Networks:
    10.0.0.0
    172.16.0.0
    192.168.1.0
Passive Interface(s):
    Serial0/0
  Routing Information Sources:
    Gateway         Distance      Last Update
    (this router)         90      20:51:48
    192.168.1.2           90      00:22:58
    172.16.10.6           90      01:58:46
    172.16.10.2           90      01:59:52
  Distance: internal 90 external 170
```

A. The metric K values don't match.

B. The AS numbers don't match.

C. There is a passive interface on RouterB.

D. There is an ACL set on RouterA.

16. How many paths will EIGRPv6 load-balance by default?

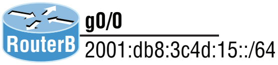
    A. 16

    B. 32

    C. 4

    D. None

17. What would your configurations be on RouterB based on the illustration? (Choose two.)



    A. `(config)#router eigrp 10`

    B. `(config)#ipv6 router eigrp 10`

    C. `(config)#ipv6 router 2001:db8:3c4d:15::/64`

    D. `(config-if)#ip eigrp 10`

    E. `(config-if)#ipv6 eigrp 10`

    F. `(config-if)#ipv6 router eigrp 10`

18. RouterA has a feasible successor not shown in the following output. Based on what you can learn from the output, which one of the following will be the successor for 2001:db8:c34d:18::/64 if the current successor fails?

```
via FE80::201:C9FF:FED0:3301 (29110112/33316), Serial0/0/0
via FE80::209:7CFF:FE51:B401 (4470112/42216), Serial0/0/1
via FE80::209:7CFF:FE51:B401 (2170112/2816), Serial0/0/2
```

    A. `Serial0/0/0`

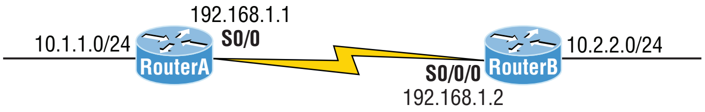
    B. `Serial0/0/1`

    C. `Serial0/0/2`

    D. There is no feasible successor.

19. You have an internetwork as shown in the following illustration with routers running IOS 12.4. However, the two networks are not sharing routing table route entries. What is the problem?

```
RouterA#sh ip protocols
Routing Protocol is "eigrp 930"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Default networks flagged in outgoing updates
  Default networks accepted from incoming updates
  EIGRP metric weight K1=1, K2=0, K3=1, K4=0, K5=0
  EIGRP maximum hopcount 100
  EIGRP maximum metric variance 2
  Redistributing: eigrp 930
  EIGRP NSF-aware route hold timer is 240s
  Automatic network summarization is in effect
  Automatic address summarization:
    192.168.1.0/24 for FastEthernet0/0
      Summarizing with metric 2169856
    10.0.0.0/8 for Serial0/0
      Summarizing with metric 28160
 [output cut]

RouterB#sh ip protocols
Routing Protocol is "eigrp 930"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Default networks flagged in outgoing updates
  Default networks accepted from incoming updates
  EIGRP metric weight K1=1, K2=0, K3=1, K4=0, K5=0
  EIGRP maximum hopcount 100
  EIGRP maximum metric variance 3
  Redistributing: eigrp 930
  EIGRP NSF-aware route hold timer is 240s
  Automatic network summarization is in effect
  Maximum path: 4
  Routing for Networks:
    10.0.0.0
    192.168.1.0
Passive Interface(s):
    Serial0/0
  Routing Information Sources:
    Gateway         Distance      Last Update
    (this router)         90      20:51:48
    192.168.1.2           90      00:22:58
    172.16.10.6           90      01:58:46
    172.16.10.2           90      01:59:52
  Distance: internal 90 external 170
```



A. The variances don't match between routers.

B. The metrics are not valid between neighbors.

C. There is a discontiguous network.

D. There is a passive interface on RouterB.

E. An ACL is set on the router.

20. Which should you look for when troubleshooting an adjacency? (Choose four.)

A. Verify the AS numbers.

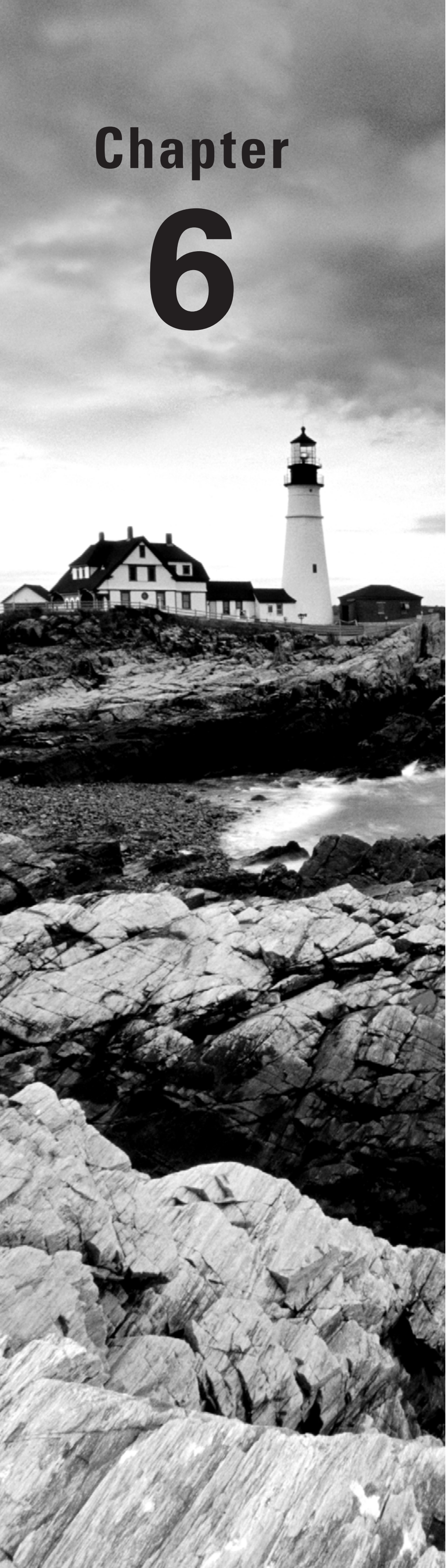
B. Verify that you have the proper interfaces enabled for EIGRP.

C. Make sure there are no mismatched K values.

D. Check your passive interface settings.

E. Make sure your remote routers are not connected to the Internet.

F. If authentication is configured, make sure all routers use different passwords.



Chapter 6

Multi-Area OSPF

THE FOLLOWING ICND2 EXAM TOPICS ARE COVERED IN THIS CHAPTER:

✓ **IP Routing Technologies**
- Configure and verify OSPF (single area)
  - Neighbor adjacencies
  - OSPF states
  - Discuss Multi-area
  - Configure OSPF v2
  - Configure OSPF v3
  - Router ID
  - LSA types

✓ **Troubleshooting**
- Troubleshoot and resolve routing issues
  - Routing is enabled
  - Routing table is correct
  - Correct path selection
- Troubleshoot and resolve OSPF problems
  - Neighbor adjacencies
  - Hello and Dead timers
  - OSPF area
  - Interface MTU
  - Network types
  - Neighbor states
  - OSPF topology database

We'll begin this chapter by focusing on the scalability constraints of an Open Shortest Path First (OSPF) network with a single area and move on from there to explore the concept of multi-area OSPF as a solution to these scalability limitations.

I'll also identify and introduce you to the various categories of routers used in multi-area configurations, including backbone routers, internal routers, area border routers (ABRs), and autonomous system boundary routers (ASBRs).

The functions of different OSPF Link-State Advertisements (LSAs) are absolutely crucial for you to understand for success in taking the Cisco exam, so I'll go into detail about the types of LSAs used by OSPF, as well as, the Hello protocol and different neighbor states when an adjacency is taking place.

And because troubleshooting is always a vital skill to have, I'll guide you through the process with a collection of `show` commands that can be effectively used to monitor and troubleshoot a multi-area OSPF implementation. Finally, I'll end the chapter with the easiest part: configuring and verifying OSPFv3.

NOTE

To find up-to-the-minute updates for this chapter, please see `www.lammle.com/forum` or the book's web page at `www.sybex.com`.

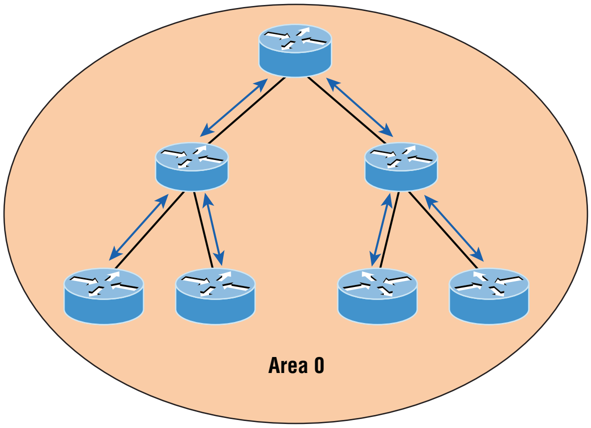
OSPF Scalability

At this point, and before you read this chapter, be sure that you have the foundation of single-area OSPF down pat. I'm sure you remember OSPF's significant advantage over distance-vector protocols like RIP, due to OSPF's ability to represent an entire network within its link-state database, which dramatically reduces the time required for convergence!

But what does a router actually go through to give us this great performance? Each router recalculates its database every time there's a topology change. If you have numerous routers in an area, they'll clearly have lots of links. Every time a link goes up or down, an LSA Type 1 packet is advertised, forcing all of the routers in the same area to recalculate their shortest path first (SPF) tree. Predictably, this kind of heavy lifting requires a ton of CPU overhead. On top of that, each router must hold the entire link-state database that represents the topology of the entire network, which results in considerable memory overhead. As if all that weren't enough, each router also holds a complete copy of the routing table, adding more to the already heavy overhead burden on memory. And keep in mind that the number of entries in the routing table can be much greater than the number of networks in the routing table because there are typically multiple routes to to the same remote networks!

Considering these OSPF factors, it's easy to imagine that in a really large network, single-area OSPF presents some serious scalability challenges, as shown in Figure 6.1. We'll move on in a bit to compare the single-area OSPF network in that illustration to our multi-area networks.

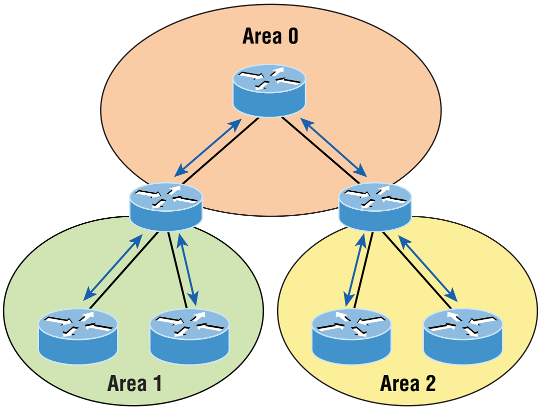
FIGURE 6.1 OSPF single-area network: All routers flood the network with link-state information to all other routers within the same area.



Single-area OSPF design places all routers into a single OSPF area, which results in many LSAs being processed on every router.

Fortunately, OSPF allows us to take a large OSPF topology and break it down into multiple, more manageable areas, as illustrated in Figure 6.2.

FIGURE 6.2 OSPF multi-area network: All routers flood the network only within their area.



Just take a minute to think about the advantages of this hierarchical approach. First, routers that are internal to a defined area don't need to worry about having a link-state database for the entire network because they need one for only their own areas. This factor seriously reduces memory overhead! Second, routers that are internal to a defined area now have to recalculate their link-state database only when there's a topology change within their given area. Topology changes in one area won't cause global OSPF recalculations, further reducing processor overhead. Finally, because routes can be summarized at area boundaries, the routing tables on each router just don't need to be nearly as huge as they would be in a single-area environment!

But of course there's a catch: As you start subdividing your OSPF topology into multiple areas, the configuration gets more complex, so we'll explore some strategic ways to finesse the configuration plus look at some cool tricks for effectively troubleshooting multi-area OSPF networks.

Categories of Multi-area Components

In the following sections, I'm going to cover the various roles that routers play in a multi-area OSPF network. You'll find routers serving as backbone routers, internal routers, area border routers, and autonomous system boundary routers. I'll also introduce you to the different types of advertisements used in an OSPF network.

Link-State Advertisements (LSAs) describe a router and the networks that are connected to it by sending the LSAs to neighbor routers. Routers exchange LSAs and learn the complete topology of the network until all routers have the exact same topology database. After the topology database is built, OSPF uses the Dijkstra algorithm to find the best path to each remote network and places only the best routes into the routing table.

Adjacency Requirements

Once neighbors have been identified, adjacencies must be established so that routing (LSA) information can be exchanged. There are two steps required to change a neighboring OSPF router into an adjacent OSPF router:

1. Two-way communication (achieved via the Hello protocol)
2. Database synchronization, which consists of three packet types being exchanged between routers:
    - Database Description (DD) packets
    - Link-State Request (LSR) packets
    - Link-State Update (LSU) packets

Once database synchronization is complete, the two routers are considered adjacent. This is how adjacency is achieved, but you need to know when an adjacency will occur.

It's important to remember that neighbors will not form an adjacency if the following do not match:

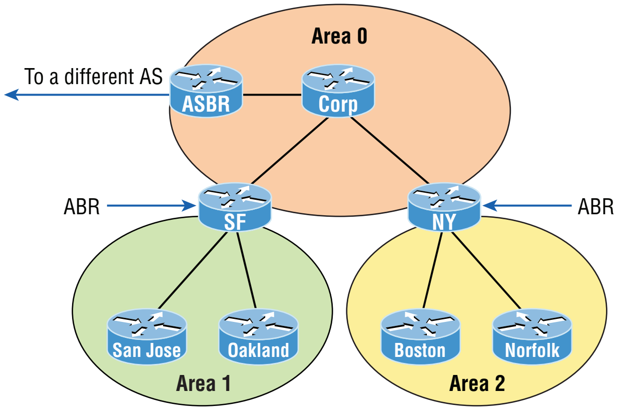
- Area ID
- Subnet
- Hello and dead timers
- Authentication (if configured)

When adjacencies form, depends on the network type. If the link is point-to-point, the two neighbors will become adjacent if the Hello packet information for both routers is configured properly. On broadcast multi-access networks, adjacencies are formed only between the OSPF routers on the network and the DR and BDR.

OSPF Router Roles

Routers within a multi-area OSPF network fall into different categories. Check out Figure 6.3 to see the various roles that routers can play.

FIGURE 6.3 Router roles: Routers within an area are called internal routers.



Notice that there are four routers that are part of area 0: the Corp router, SF and NY, and the autonomous system border router (ASBR). When configuring multi-area OSPF, one area must be called area 0, referred to as the *backbone area*. All other areas must connect to area 0. The four routers are referred to as the backbone routers, which are any routers that exist either partially or completely in OSPF area 0.

Another key distinction about the SF and NY routers connecting to other areas is that they have interfaces in more than one area. This makes them *area border routers (ABRs)* because in addition to having an interface in area 0, SF has an interface in area 1 and NY has an interface in area 2.

An ABR is a router that belongs to more than one OSPF area. It maintains information from all directly connected areas in its topology table but doesn't share the topological details from one area with the other. But it will forward routing information from one area to the other. The key concept here is that an ABR separates the LSA flooding zone, is a primary point for area address summarization, and typically has the source default route, all while maintaining the link-state database (LSDB) for each area it's connected to.

TIP Remember that a router can play more than one role. In Figure 6.3, SF and NY are both backbone routers and area border routers.

Let's turn our focus to the San Jose and Oakland routers. You can see that all interfaces on both of these routers reside only in area 1. Because all of San Jose's and Oakland's interfaces are internal to a single area, they're called internal routers. An *internal router* is any router with all of its interfaces included as members of the same area. This also applies to the Boston and Norfolk routers and their relationship to area 2. The Corp router is internal to area 0.

Finally, the ASBR is unique among all routers in our example because of its connection to an external *autonomous system (AS)*. When an OSPF network is connected to an EIGRP network, a *Border Gateway Protocol (BGP)* network, or a network running any other external routing process, it's referred to as an AS.

An *autonomous system boundary router (ASBR)* is an OSPF router with at least one interface connected to an external network or different AS. A network is considered external if it's either running a routing protocol other than OSPF. An ASBR is responsible for injecting route information learned via the external network into OSPF.

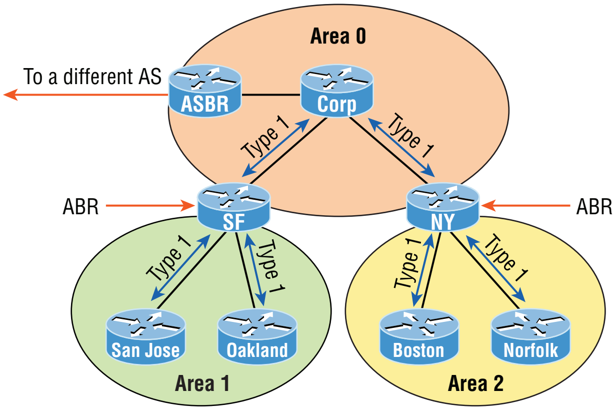
I want to point out that an ASBR doesn't automatically exchange routing information between its OSPF routing process and the external routing process that it's connected to. These routes are exchanged through a method called *route redistribution*, which is beyond the scope of this book.

Link-state Advertisements

You know that a router's link-state database is made up of *Link-State Advertisements (LSAs)*. But just as there are several OSPF router categories to remember, there are also various types of LSAs to keep in mind—five of them, to be exact. These LSA classifications may not seem important at first, but you'll see why they are when we cover how the various types of OSPF areas operate. Let's start by exploring the different types of LSAs that Cisco uses:

Type 1 LSA Referred to as a *router link advertisement (RLA)*, or just router LSA, a *Type 1 LSA* is sent by every router to other routers in its area. This advertisement contains the status of a router's link in the area to which it is connected. If a router is connected to multiple areas, then it will send separate Type 1 LSAs for each of the areas it's connected to. Type 1 LSAs contain the router ID (RID), interfaces, IP information, and current interface state. For example, in the network in Figure 6.4, router SF will send an LSA Type 1 advertisement for its interface into area 0 and a separate LSA Type 1 advertisement for its interfaces into area 1 describing the state of its links. The same will happen with the other routers in Figure 6.4.

FIGURE 6.4 Type 1 Link-state Advertisements



Type 1: Here is the status of my links!

Type 2 LSA Referred to as a *network link advertisement (NLA)*, a *Type 2 LSA* is generated by designated routers (DRs). Remember that a designated router is elected to represent other routers in its network, and it establishes adjacencies with them. The DR uses a Type 2 LSA to send out information about the state of other routers that are part of the same network. Note that the Type 2 LSA is flooded to all routers that are in the same area as the one containing the specific network but not to any outside of that area. These updates contain the DR and BDR IP information.

Type 3 LSA Referred to as a *summary link advertisement (SLA)*, a *Type 3 LSA* is generated by area border routers. These ABRs send Type 3 LSAs toward the area external to the one where they were generated. The Type 3 LSA advertises networks, and these LSAs advertise *inter-area routes* to the backbone area (area 0). Advertisements contain the IP information and RID of the ABR that is advertising an LSA Type 3.

NOTE

The word *summary* often invokes images of a summarized network address that hides the details of many small subnets within the advertisement of a single large one. But in OSPF, Summary Link Advertisements don't necessarily contain network summaries. Unless the administrator manually creates a summary, the full list of individual networks available within an area will be advertised by the SLAs.

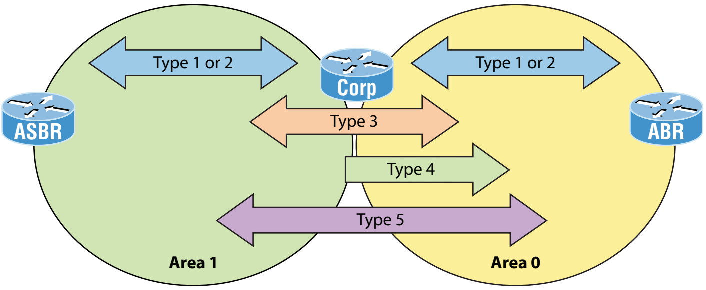
Type 4 LSA *Type 4 LSAs* are generated by area border routers. These ABRs send a Type 4 LSA toward the area external to the one in which they were generated. These are also summary LSAs like Type 3, but Type 4 are specifically used to inform the rest of the OSPF areas how to get to the ASBR.

Type 5 LSA Referred to as *AS external link advertisements*, a *Type 5 LSA* is sent by autonomous system boundary routers to advertise routes that are external to the OSPF autonomous system and are flooded everywhere. A Type 5 LSA is generated for each individual external network advertised by the ASBR.

Figure 6.5 shows how each LSA type would be used in a multi-area OSPF network.

FIGURE 6.5 Basic LSA types

LSA Type	Description
1	Router LSA
2	Network LSA
3	Summary LSA
4	ASBR Summary LSA
5	Autonomous system LSA



It's important to understand the different LSA types and how they work. Looking at Figure 6.5, you can see that Type 1 and 2 are flooded between routers in their same area. Type 3 LSAs from the Corp router (which is an ABR and maintains the LSDB for each area it is connected to) will summarize information learned from area 1 into area 0 and vice versa. The ASBR will flood Type 5 LSAs into area 1, and the Corp router will then flood Type 4 LSAs into area 0, telling all routers how to get to the ASBR, basically becoming a proxy ASBR.

OSPF Hello Protocol

The Hello protocol provides a lot of information to neighbors. The following is communicated between neighbors, by default, every 10 seconds:

Router ID (RID) This is the highest active IP address on the router. The highest loopback IP addresses are used first. If no loopback interfaces are configured, OSPF will choose from physical interfaces instead.

Hello/Dead interval The period between Hello packets is the Hello time, which is 10 seconds by default. The dead time is the length of time allotted for a Hello packet to be received before a neighbor is considered down—four times the Hello interval, unless otherwise configured.

Neighbors The information includes a list of the router IDs for all the originating router's neighbors; neighbors being defined as routers that are attached to a common IP subnet and use identical subnet masks.

Area ID This represents the area that the originating router interface belongs to.

Router priority The priority is an 8-bit value used to aid in the election of the DR and BDR. This isn't set on point-to-point links!

DR IP address This is the router ID of the current DR.

BDR IP address This is the router ID of the current BDR.

Authentication data This is the authentication type and corresponding information (if configured).

The mandatory information within the Hello update that must match exactly are the hello and dead timer values intervals, area ID, OSPF area type, subnet, and authentication data if used. If any of those don't match perfectly, no adjacency will occur!

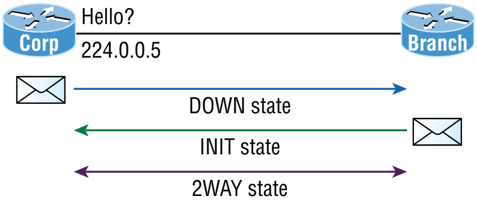
Neighbor States

Before we move on to configuration, verification, and troubleshooting OSPF, it's important for you to grasp how OSPF routers traverse different states when adjacencies are being established.

When OSPF routers are initialized, they first start exchanging information using the Hello protocol via the multicast address 224.0.0.5. After the neighbor relationship is established between routers, the routers synchronize their link-state database (LSDB) by reliably exchanging LSAs. They actually exchange quite a bit of vital information when they start up.

The relationship that one router has with another consists of eight possible states. All OSPF routers begin in the DOWN state, and if all is well, they'll progress to either the 2WAY or FULL state with their neighbors. Figure 6.6 shows this neighbor state progression.

FIGURE 6.6 OSPF neighbor states, part 1



The process starts by sending out Hello packets. Every listening router will then add the originating router to the neighbor database. The responding routers will reply with all of their Hello information so that the originating router can add them to its own neighbor table. At this point, we will have reached the 2WAY state—only certain routers will advance beyond this to establish adjacencies.

Here's a definition of the eight possible relationship states:

DOWN In the *DOWN state*, no Hello packets have been received on the interface. Bear in mind that this does not imply that the interface itself is physically down.

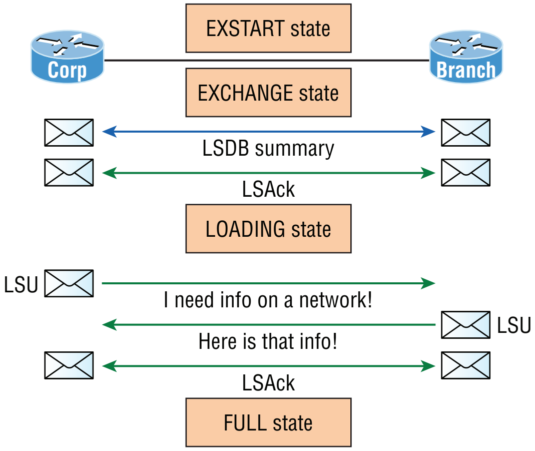
ATTEMPT In the *ATTEMPT state*, neighbors must be configured manually. It applies only to nonbroadcast multi-access (NBMA) network connections.

INIT In the *INIT state*, Hello packets have been received from another router. Still, the absence of the Router ID for the receiving router in the Neighbor field indicates that bidirectional communication hasn't been established yet.

2WAY In the *2WAY state*, Hello packets that include their own router ID in the Neighbor field have been received. Bidirectional communication has been established. In broadcast multi-access networks, an election can occur after this point.

After the DR and BDR have been selected, the routers will enter into the EXSTART state and the routers are ready to discover the link-state information about the internetwork and create their LSDB. This process is illustrated in Figure 6.7.

FIGURE 6.7 OSPF router neighbor states, part 2



EXSTART In the *EXSTART state*, the DR and BDR establish adjacencies with each router in the network. A master-slave relationship is created between each router and its adjacent DR and DBR. The router with the highest RID becomes the master, and the master-slave election dictates which router will start the exchange. Once routers exchange DBD packets, the routers will move into the EXCHANGE state.

EXCHANGE In the *EXCHANGE state*, routing information is exchanged using Database Description (DBD or DD) packets and Link-State Request (LSR) and Link-State Update packets may also be sent. When routers start sending LSRs, they're considered to be in the LOADING state.

LOADING In the *LOADING state*, Link-State Request (LSR) packets are sent to neighbors to request any Link-State Advertisements (LSAs) that may have been missed or corrupted while the routers were in the EXCHANGE state. Neighbors respond with Link-State Update (LSU) packets, which are in turn acknowledged with Link-State Acknowledgement (LSAck) packets. When all LSRs have been satisfied for a given router, the adjacent routers are considered synchronized and enter the FULL state.

FULL In the *FULL state*, all LSA information is synchronized among neighbors and adjacency has been established. OSPF routing can begin only after the FULL state has been reached!

It's important to understand that routers should be in the 2WAY and FULL states and the others are considered transitory. Routers shouldn't remain in any other state for extended period of times. Let's configure OSPF now to see what we've covered so far in action.

Basic Multi-area Configuration

Visit `ccna.gg/ch6/a` for a companion MicroNugget from CBT Nuggets.

Basic multi-area configuration isn't all that hard. Understanding your design, layout, types of LSAs, DRs and configuring the elections, troubleshooting, and fully comprehending what's happening in the background are really the most complicated aspects of OSPF.

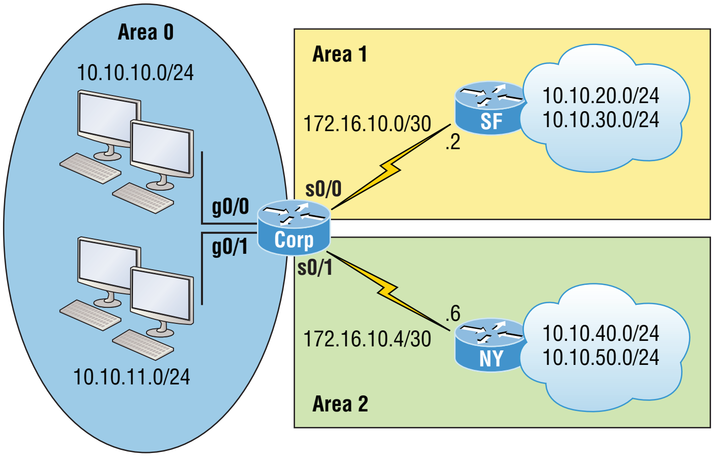
As I was saying, configuring OSPF is pretty simple, and you'll see toward the end of this chapter that configuring OSPFv3 is even easier! After I show you the basic OSPF multi-area configuration in this section, we'll work on the verification of OSPF and then go through a detailed troubleshooting scenario just as we did with EIGRP. Let's get the ball rolling with the multi-area configuration shown in Figure 6.8.

We'll use the same routers we've been working with throughout all the chapters, but we're going to create three areas. The routers are still configured with the IPv6 addresses from my last EIGRPv6 section in the previous chapter, and I've also verified that the IPv4 addresses are on the interfaces and working as well since then, so we're all set to rock the configs for this chapter! Here's the Corp configuration:

```
Corp#config t
Corp(config)#router ospf 1
Corp(config-router)#router-id 1.1.1.1
Corp(config-router)#Reload or use "clear ip ospf process" command, for this to
take effect
```

```
Corp(config-router)#network 10.10.0.0 0.0.255.255 area 0
Corp(config-router)#network 172.16.10.0 0.0.0.3 area 1
Corp(config-router)#network 172.16.10.4 0.0.0.3 area 2
```

FIGURE 6.8 Our internetwork



Pretty straightforward, but let's talk about it anyway. First I started the OSPF process with the router `ospf process-id` command, using any number from 1–65,535 because they're only locally significant so they don't need to match my neighbor routers. I set the RID of the router only to remind you that this can be configured under the router process, but with our small network it wouldn't really be necessary to mess with RIDs if this was an actual production network. The one thing that you need to keep in mind here is that in OSPF, the RID must be different on each router. With EIGRP, they can all be the same because they are not as important in that process. Still, as I showed you in the EIGRPv6 section, we still need them!

Anyway, at this point in the configurations I needed to choose my network statements for the OSPF process to use, which allowed me to place my four interfaces on the Corp router into three different areas. In the first network statement, 10.10.0.0 0.0.255.255, placed the gi0/0 and gi0/1 interfaces into area 0. The second and third statements needed to be more exact since there are /30 networks. 172.16.10.0 0.0.0.3 tells OSPF process 1 to go find an active interface that's configured with 172.16.10.1 or .2 and to place that interface into area 1. The last line tells the OSPF process to go find any active interface configured with 172.16.10.5 or .6 and place that interface into area 2. The wildcard of 0.0.0.3 means the first three octets can match any value, but the last octet is a block size of 4.

The only thing different about these configurations from those in the single-area OSPF is the different areas at the end of the command—that's it!

Here is the configuration for the SF and NY routers:

```
SF(config)#router ospf 1
SF(config-router)#network 10.10.0.0 0.0.255.255 area 1
SF(config-router)#network 172.16.0.0 0.0.255.255 area 1


NY(config)#router ospf 1
NY(config-router)#network 0.0.0.0 255.255.255.255 area 2
00:01:07: %OSPF-5-ADJCHG: Process 1, Nbr 1.1.1.1 on Serial0/0/0 from LOADING to
FULL,
Loading Done
```

I configured each one slightly different from the Corp router, but since they didn't have an interface in more than area 1, I had more leeway in configuring them. For the NY router I just configured a network statement (0.0.0.0 255.255.255.255) that says "go find any active interface and place it into area 2!" I'm not recommending that you configure your routers in such a broad manner; I just wanted to show you your options.

Now that our three routers are configured, let's verify our internetwork.

Visit `ccna.gg/ch6/b` for a companion MicroNugget from CBT Nuggets.

Verifying and Troubleshooting Multi-area OSPF Networks

Cisco's IOS has several `show` and `debug` commands that can help you monitor and troubleshoot OSPF networks. A sampling of these commands, which can be used to gain information about various OSPF characteristics, is included in Table 6.1.

TABLE 6.1 OSPF verification commands

Command	Provides the following
`show ip ospf neighbor`	Verifies your OSPF-enabled interfaces
`show ip ospf interface`	Displays OSPF-related information on an OSPF-enabled interface
`show ip protocols`	Verifies the OSPF process ID and that OSPF is enabled on the router
`show ip route`	Verifies the routing table, and displays any OSPF injected routes
`show ip ospf database`	Lists a summary of the LSAs in the database, with one line of output per LSA, organized by type

Let's go through some verification commands—the same commands we used to verify our single-area OSPF network—then we'll move onto the OSPF troubleshooting scenario section.

Okay, once you've checked the link between your neighbors and can use the Ping program, the best command when verifying a routing protocol is to always check the status of your neighbor's connection first. The `show ip ospf neighbor` command is super useful because it summarizes the pertinent OSPF information regarding neighbors and their adjacency state. If a DR or BDR exists, that information will also be displayed. Here's a sample:

```
Corp#sh ip ospf neighbor
Neighbor ID     Pri   State           Dead Time   Address         Interface
172.16.10.2       0   FULL/  -        00:00:34    172.16.10.2     Serial0/0/0
172.16.10.6       0   FULL/  -        00:00:31    172.16.10.6     Serial0/0/1


SF#sh ip ospf neighbor
Neighbor ID     Pri   State           Dead Time   Address         Interface
1.1.1.1           0   FULL/  -        00:00:39    172.16.10.1     Serial0/0/0


NY#sh ip ospf neighbor
Neighbor ID     Pri   State           Dead Time   Address         Interface
1.1.1.1           0   FULL/  -        00:00:34    172.16.10.5     Serial0/0/0
```

The reason that the Corp connections to SF and LA don't have a DR or BDR listed in the output is that by default, elections don't happen on point-to-point links and they show `FULL/-`. But we can see that the Corp router is fully adjacent to all three routers from its output.

The output of this command shows the neighbor ID, which is the RID of the router. Notice in the output of the Corp router that the RIDs for the SF and NY routers were chosen based on highest IP address of any active interface when I started the OSPF process on those routers. Both the SF and NY routers see the Corp router RID as 1.1.1.1 because I set that manually under the `router ospf` process command.

Next we see the `Pri` field, which is the priority field that's set to 1 by default. Don't forget that on point-to-point links, elections don't happen, so the interfaces are all set to 0 in this example because none of these routers will have elections on these interfaces with each other over this serial WAN network. The state field shows `Full/-`, which means all routers are synchronized with their LSDB, and the /- means there is no election on this type of interface. The dead time is counting down, and if the router does not hear from this neighbor before this expires, the link will be considered down. The address is the actual address of the neighbor's interface connecting to the router, and the interface is the neighbor's interface—not your interface!

The *show ip ospf* Command

We use the `show ip ospf` command to display OSPF information for one or all OSPF processes running on the router. Information contained therein includes the router ID, area information, SPF statistics, and LSA timer information. Let's check out the output from the Corp router:

```
Corp#sh ip ospf
 Routing Process "ospf 1" with ID 1.1.1.1
 Supports only single TOS(TOS0) routes
 Supports opaque LSA
 It is an area border router
 SPF schedule delay 5 secs, Hold time between two SPFs 10 secs
 Minimum LSA interval 5 secs. Minimum LSA arrival 1 secs
 Number of external LSA 0. Checksum Sum 0x000000
 Number of opaque AS LSA 0. Checksum Sum 0x000000
 Number of DCbitless external and opaque AS LSA 0
 Number of DoNotAge external and opaque AS LSA 0
 Number of areas in this router is 3. 3 normal 0 stub 0 nssa
 External flood list length 0
    Area BACKBONE(0)
        Number of interfaces in this area is 2
        Area has no authentication
        SPF algorithm executed 19 times
        Area ranges are
        Number of LSA 7. Checksum Sum 0x0384d5
        Number of opaque link LSA 0. Checksum Sum 0x000000
        Number of DCbitless LSA 0
        Number of indication LSA 0
        Number of DoNotAge LSA 0
        Flood list length 0
    Area 1
        Number of interfaces in this area is 1
        Area has no authentication
        SPF algorithm executed 43 times
        Area ranges are
        Number of LSA 7. Checksum Sum 0x0435f8
        Number of opaque link LSA 0. Checksum Sum 0x000000
        Number of DCbitless LSA 0
        Number of indication LSA 0
        Number of DoNotAge LSA 0
        Flood list length 0
    Area 2
        Number of interfaces in this area is 1
        Area has no authentication
        SPF algorithm executed 38 times
        Area ranges are
        Number of LSA 7. Checksum Sum 0x0319ed
        Number of opaque link LSA 0. Checksum Sum 0x000000
        Number of DCbitless LSA 0
        Number of indication LSA 0
        Number of DoNotAge LSA 0
        Flood list length 0
```

You'll notice that most of the preceding information wasn't displayed with this command output in single-area OSPF. We have more displayed here because it's providing information about each area we've configured on this router.

The *show ip ospf interface* Command

The `show ip ospf interface` command displays all interface-related OSPF information. Data is displayed for all OSPF-enabled interfaces or for specified interfaces. I'll highlight some important portions I want you to pay special attention to.

```
Corp#sh ip ospf interface gi0/0
GigabitEthernet0/0 is up, line protocol is up
  Internet address is 10.10.10.1/24, Area 0
  Process ID 1, Router ID 1.1.1.1, Network Type BROADCAST, Cost: 1
  Transmit Delay is 1 sec, State DR, Priority 1
  Designated Router (ID) 1.1.1.1, Interface address 10.10.10.1
  No backup designated router on this network
  Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
    Hello due in 00:00:05
  Index 1/1, flood queue length 0
  Next 0x0(0)/0x0(0)
  Last flood scan length is 1, maximum is 1
  Last flood scan time is 0 msec, maximum is 0 msec
  Neighbor Count is 0, Adjacent neighbor count is 0
  Suppress hello for 0 neighbor(s)
```

Let's take a look at a serial interface so we can compare it to the Gigabit Ethernet interface just shown. The Ethernet network is a broadcast multi-access network by default, and the serial interface is a point-to-point nonbroadcast multi-access network, so they will act differently with OSPF:

```
Corp#sh ip ospf interface s0/0/0
Serial0/0/0 is up, line protocol is up
  Internet address is 172.16.10.1/30, Area 1
  Process ID 1, Router ID 1.1.1.1, Network Type POINT-TO-POINT, Cost: 64
  Transmit Delay is 1 sec, State POINT-TO-POINT, Priority 0
  No designated router on this network
  No backup designated router on this network
  Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
    Hello due in 00:00:02
  Index 3/3, flood queue length 0
  Next 0x0(0)/0x0(0)
  Last flood scan length is 1, maximum is 1
  Last flood scan time is 0 msec, maximum is 0 msec
  Neighbor Count is 1 , Adjacent neighbor count is 1
    Adjacent with neighbor 172.16.10.2
  Suppress hello for 0 neighbor(s)
```

The following information is displayed via this command:

- Interface IP address
- Area assignment
- Process ID
- Router ID
- Network type
- Cost
- Priority
- DR/BDR election information (if applicable)
- Hello and dead timer intervals
- Adjacent neighbor information

I used the `show ip ospf interface gi0/0` command first because I knew that there would be a designated router elected on the Ethernet broadcast multi-access network on the Corp router, even though it has no one to run against, which means the Corp router automatically wins. The information that I bolded is all very important! What are the hello and dead timers set to by default? Even though I haven't talked much about the cost output on an interface, it can also be very important. Two OSPF routers still could create an adjacency if the costs don't match, but it could lead to certain links not being utilized. We'll discuss this more at the end of the verification section.

Real World Scenario

Neighbor Routers Don’t Form an Adjacency

I’d like to talk more about the adjacency issue and how the `show ip ospf interface` command can help you solve problems, especially in multi-vendor networks.

Years ago I was consulting with the folks at a large PC/laptop manufacturer and was helping them build out their large internetwork. They were using OSPF because their company was a worldwide company and used many types of routers from all manufacturers.

I received a call from a remote branch informing me that they installed a new router but it was not seeing the Cisco router off their Ethernet interface. Of course it was an emergency because this new router was holding some important WAN links to a new remote location that needed to be up yesterday!

After calming down the person on the phone, I simply had the admin use the `show ip ospf interface fa0/0` command and verify the hello and dead timers and the area configured for that interface and then had him verify that the IP addresses were correct between routers and that there was no passive interface set.

Then I had him verify that same information on the neighbor, and sure enough the neighbors’ hello and dead timers didn’t match. Quick and easy fix on the interface of the Cisco router with the `ip ospf dead 30` command, and they were up!

Always remember that OSPF can work with multi-vendor routers, but no one ever said it works out of the box between various vendors!

The *show ip protocols* Command

The `show ip protocols` command is also useful, whether you’re running OSPF, EIGRP, RIP, BGP, IS-IS, or any other routing protocol that can be configured on your router. It provides an excellent overview of the actual operation of all currently running protocols.

Check the output from the Corp router:

```
Corp#sh ip protocols
Routing Protocol is "ospf 1"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Router ID 1.1.1.1
  Number of areas in this router is 3. 3 normal 0 stub 0 nssa
  Maximum path: 4
  Routing for Networks:
    10.10.0.0 0.0.255.255 area 0
    172.16.10.0 0.0.0.3 area 1
    172.16.10.4 0.0.0.3 area 2
  Routing Information Sources:
    Gateway         Distance      Last Update
    1.1.1.1              110      00:17:42
    172.16.10.2          110      00:17:42
    172.16.10.6          110      00:17:42
  Distance: (default is 110)
```

Here we can determine the OSPF process ID, OSPF router ID, type of OSPF area, networks, and the three areas configured for OSPF, as well as, the OSPF router IDs of neighbors—that's a lot. Read efficient!

The *show ip route* Command

Now would be a great time to issue a `show ip route` command on the Corp router. The Corp router shows only four dynamic routes for our internetwork, with the *O* representing OSPF internal routes. The *C*s clearly represent our directly connected networks, but our four remote networks are also showing up—nice! Notice the 110/65, which is the administrative distance/metric:

```
Corp#sh ip route
[output cut]
     10.0.0.0/8 is variably subnetted, 8 subnets, 2 masks
C       10.10.10.0/24 is directly connected, GigabitEthernet0/0
L       10.10.10.1/32 is directly connected, GigabitEthernet0/0
C       10.10.11.0/24 is directly connected, GigabitEthernet0/1
L       10.10.11.1/32 is directly connected, GigabitEthernet0/1
O       10.10.20.0/24 [110/65] via 172.16.10.2, 02:18:27, Serial0/0/0
O       10.10.30.0/24 [110/65] via 172.16.10.2, 02:18:27, Serial0/0/0
O       10.10.40.0/24 [110/65] via 172.16.10.6, 03:37:24, Serial0/0/1
O       10.10.50.0/24 [110/65] via 172.16.10.6, 03:37:24, Serial0/0/1
     172.16.0.0/16 is variably subnetted, 4 subnets, 2 masks
C       172.16.10.0/30 is directly connected, Serial0/0/0
L       172.16.10.1/32 is directly connected, Serial0/0/0
C       172.16.10.4/30 is directly connected, Serial0/0/1
L       172.16.10.5/32 is directly connected, Serial0/0/1
```

In addition, you can use the `show ip route ospf` command to get only OSPF-injected routes in your routing table. I can't stress enough how useful this is when dealing with large networks!

```
Corp#sh ip route ospf
     10.0.0.0/8 is variably subnetted, 8 subnets, 2 masks
O       10.10.20.0 [110/65] via 172.16.10.2, 02:18:33, Serial0/0/0
O       10.10.30.0 [110/65] via 172.16.10.2, 02:18:33, Serial0/0/0
O       10.10.40.0 [110/65] via 172.16.10.6, 03:37:30, Serial0/0/1
O       10.10.50.0 [110/65] via 172.16.10.6, 03:37:30, Serial0/0/1
```

Now that's a really nice-looking OSPF routing table! Troubleshooting and fixing an OSPF network is as vital a skill to have as it is in any other networking environment, which is why I always use the `show ip int brief` command when configuring my routing protocols. It's very easy to make little mistakes with OSPF, so pay very close attention to the details—especially when troubleshooting!

The *show ip ospf database* Command

Using the `show ip ospf database` command will give you information about the number of routers in the internetwork (AS), plus the neighboring router's ID. This is the topology database I referred to earlier.

The output is broken down by area. Here's a sample, again from Corp:

```
Corp#sh ip ospf database
            OSPF Router with ID (1.1.1.1) (Process ID 1)


                Router Link States (Area 0)


Link ID         ADV Router      Age         Seq#       Checksum Link count
1.1.1.1         1.1.1.1         196         0x8000001a 0x006d76 2


                Summary Net Link States (Area 0)
Link ID         ADV Router      Age         Seq#       Checksum
172.16.10.0     1.1.1.1         182         0x80000095 0x00be04
172.16.10.4     1.1.1.1         177         0x80000096 0x009429
10.10.40.0      1.1.1.1         1166        0x80000091 0x00222b
10.10.50.0      1.1.1.1         1166        0x80000092 0x00b190
10.10.20.0      1.1.1.1         1114        0x80000093 0x00fa64
10.10.30.0      1.1.1.1         1114        0x80000094 0x008ac9
```

```
                Router Link States (Area 1)

Link ID         ADV Router      Age         Seq#       Checksum Link count
1.1.1.1         1.1.1.1         1118        0x8000002a 0x00a59a 2
172.16.10.2     172.16.10.2     1119        0x80000031 0x00af47 4

                Summary Net Link States (Area 1)
Link ID         ADV Router      Age         Seq#       Checksum
10.10.10.0      1.1.1.1         178         0x80000076 0x0021a5
10.10.11.0      1.1.1.1         178         0x80000077 0x0014b0
172.16.10.4     1.1.1.1         173         0x80000078 0x00d00b
10.10.40.0      1.1.1.1         1164        0x80000074 0x005c0e
10.10.50.0      1.1.1.1         1164        0x80000075 0x00eb73

                Router Link States (Area 2)

Link ID         ADV Router      Age         Seq#       Checksum Link count
1.1.1.1         1.1.1.1         1119        0x8000002b 0x005cd6 2
172.16.10.6     172.16.10.6     1119        0x8000002d 0x0020a3 4

                Summary Net Link States (Area 2)
Link ID         ADV Router      Age         Seq#       Checksum
10.10.10.0      1.1.1.1         179         0x8000007a 0x0019a9
10.10.11.0      1.1.1.1         179         0x8000007b 0x000cb4
172.16.10.0     1.1.1.1         179         0x8000007c 0x00f0ea
10.10.20.0      1.1.1.1         1104        0x80000078 0x003149
10.10.30.0      1.1.1.1         1104        0x80000079 0x00c0ae
Corp#
```

Considering we only have eight networks configured in our internetwork, there's a huge amount of information in this database! You can see all the routers and the RID of each—the highest IP address related to individual routers. And each output under each area represents LSA Type 1, indicating the area they're connected to.

The router output also shows the link ID. Remember that an interface is also a link, as is the RID of the router on that link under the ADV router—the advertising router.

So far, this has been a great chapter, brimming with detailed OSPF information, a whole lot more than what was needed to meet past Cisco objectives, for sure! Next, we'll use the same sample network that I built in Chapter 5 on EIGRP and run through a troubleshooting scenario using multi-area OSPF.

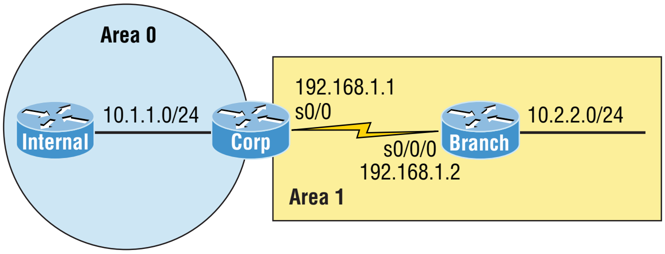
Troubleshooting OSPF Scenario

When you notice problems with your OSPF network, it's wise to first test your layer 3 connectivity with Ping and the `traceroute` command to see if your issue is a local one. If all looks good locally, then follow these Cisco-provided guidelines:

1. Verify your adjacency with your neighbor routers using the `show ip ospf neighbors` command. If you are not seeing your neighbor adjacencies, then you need to verify that the interfaces are operational and enabled for OSPF. If all is well with the interfaces, verify the hello and dead timers next, and establish that the interfaces are in the same area and that you don't have a passive interface configured.
2. Once you've determined that your adjacencies to all neighbors are working, use the `show ip route` to verify your layer 3 routes to all remote networks. If you see no OSPF routes in the routing table, you need to verify that you don't have another routing protocol running with a lower administrative distance. You can use `show ip protocols` to see all routing protocols running on your router. If no other protocols are running, then verify your `network` statements under the OSPF process. In a multi-area network, make sure all non-backbone area routers are directly connected to area 0 through an ABR or they won't be able to send and receive updates.
3. If you can see all the remote networks in the routing table, move on to verify the path for each network and that each path for specific networks is correct. If not, you need to verify the cost on your interfaces with the `show ip ospf interface` command. You may need to adjust the cost on an interface either higher or lower, depending on which path you want OSPF to use for sending packets to a remote network. Remember—the path with the lowest cost is the preferred path!

Okay, with our marching orders for troubleshooting OSPF in hand, let's take a look at Figure 6.9, which we'll use to verify our network now.

FIGURE 6.9 Our internetwork



Here's the OSPF configuration on the three routers:

```
Corp(config-if)#router ospf 1
Corp(config-router)#network 10.1.1.0 0.0.0.255 area 0
Corp(config-router)#network 192.168.1.0 0.0.0.3 area 1
```

```
Internal(config)#router ospf 3
Internal(config-router)#network 10.1.1.2 0.0.0.0 area 0
```

```
Branch(config-if)#router ospf 2
Branch(config-router)#network 192.168.1.2 0.0.0.0 area 0
Branch(config-router)#network 10.2.2.1 0.0.0.0 area 0
```

Let’s check out our network now, beginning by checking the layer 1 and layer 2 status between routers:

```
Corp#sh ip int brief
Interface               IP-Address      OK? Method Status           Protocol
FastEthernet0/0         10.1.1.1        YES manual up                   up
Serial0/0               192.168.1.1     YES manual up                   up
```

The IP addresses look correct and the layer 1 and 2 status is up/up, so next we’ll use the Ping program to check connectivity like this:

```
Corp#ping 192.168.1.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms
Corp#ping 10.1.1.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.1.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms
```

Nice—I can ping both directly connected neighbors, so this means layers 1, 2, and 3 are working between neighbor routers. This is a great start, but it still doesn’t mean OSPF is actually working yet. If any of the preceding commands had failed, I first would’ve verified layers 1 and 2 to make sure my data link was working between neighbors and then moved on to verify my layer 3 IP configuration.

Since our data link appears to be working between each neighbor, our next move is to check the OSPF configuration and status of the routing protocol. I’ll start with the interfaces:

```
Corp#sh ip ospf interface s0/0
Serial0/0 is up, line protocol is up
  Internet Address 192.168.1.1/30, Area 1
  Process ID 1, Router ID 192.168.1.1, Network Type POINT_TO_POINT, Cost: 100
  Transmit Delay is 1 sec, State POINT_TO_POINT
  Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
    oob-resync timeout 40
    Hello due in 00:00:03
  Supports Link-local Signaling (LLS)
  Cisco NSF helper support enabled
  IETF NSF helper support enabled
  Index 1/2, flood queue length 0
  Next 0x0(0)/0x0(0)
  Last flood scan length is 1, maximum is 1
  Last flood scan time is 0 msec, maximum is 0 msec
  Neighbor Count is 1, Adjacent neighbor count is 1
    Adjacent with neighbor 192.168.1.2
  Suppress hello for 0 neighbor(s)
```

I've highlighted the important statistics that you should always check first on an OSPF interface. You need to verify that the interface is configured in the same area as the neighbor and that the hello and dead timers match. A cost mismatch won't stop an adjacency from forming, but it could cause ugly routing issues. We'll explore that more in a minute.

For now let's take a look at the LAN interface that's connecting to the Internal router:

```
Corp#sh ip ospf int f0/0
FastEthernet0/0 is up, line protocol is up
  Internet Address 10.1.1.1/24, Area 0
  Process ID 1, Router ID 192.168.1.1, Network Type BROADCAST, Cost: 1
  Transmit Delay is 1 sec, State DR, Priority 1
  Designated Router (ID) 192.168.1.1, Interface address 10.1.1.1
  Backup Designated router (ID) 10.1.1.2, Interface address 10.1.1.2
  Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
    oob-resync timeout 40
    Hello due in 00:00:00
  Supports Link-local Signaling (LLS)
  Cisco NSF helper support enabled
  IETF NSF helper support enabled
  Index 1/1, flood queue length 0
  Next 0x0(0)/0x0(0)
  Last flood scan length is 1, maximum is 1
  Last flood scan time is 0 msec, maximum is 0 msec
  Neighbor Count is 1, Adjacent neighbor count is 1
    Adjacent with neighbor 10.1.1.2  (Backup Designated Router)
  Suppress hello for 0 neighbor(s)
```

We'll focus on the same key factors on a LAN interface that we did on our serial interface: the area ID and hello and dead timers. Notice that the cost is 1. According to Cisco's method of calculating cost, anything 100 Mbps or higher will always be a cost of 1 and serial links with the default bandwidth are always 64. This can cause problems in a large network with lots of high-bandwidth links. One thing to take special note of is that there's a designated and backup designated router on a broadcast multi-access network. DRs and BDRs won't cause a routing problem between neighbors, but it's still a consideration when designing and configuring in a really large internetwork environment. But we won't be focusing on that for our purposes here. It's just something to keep in mind.

Staying with the troubleshooting step of checking our interfaces, look at the error I received when I tried to verify OSPF on the fa0/1 interface of the Corp router (which we're not using):

```
Corp#sh ip ospf int fa0/1
%OSPF: OSPF not enabled on FastEthernet0/1
Corp#
```

I got this error because the network statements under the OSPF process are not enabled for the network on the fa0/1 interface. If you receive this error, immediately check your network statements!

Next, let's check out the networks we're routing for with the `show ip protocols` command:

```
Corp#sh ip protocols
Routing Protocol is "ospf 1"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Router ID 192.168.1.1
  It is an area border router
  Number of areas in this router is 2. 2 normal 0 stub 0 nssa
  Maximum path: 4
  Routing for Networks:
    10.1.1.0 0.0.0.255 area 0
    192.168.1.0 0.0.0.3 area 1
 Reference bandwidth unit is 100 mbps
  Routing Information Sources:
    Gateway         Distance      Last Update
    192.168.1.2          110      00:28:40
  Distance: (default is 110)
```

From this output we can check our process ID, as well as, reveal if we have an ACL set on our routing protocol, just as we found when troubleshooting EIGRP in the last chapter. But this time, we'll first examine the network statements and the area they're configured for—most important, the specific areas that each interface is configured for. This is key because if your neighbor's interface isn't in the same area, you won't be able to form an adjacency! This command's output provides a great view of what exactly we typed in for the network statements under the OSPF process. Also, notice that the default reference bandwidth is set to 100 Mbps. I'll talk about this factor at the end of this section.

I want to point out that the neighbor IP address and administrative distance is listed. OSPF uses 110 by default, so remember that if EIGRP were running here, we wouldn't see OSPF routes in the routing table because EIGRP has an AD of 90!

Next, we'll look at our neighbor table on the Corp router to find out if OSPF has formed an adjacency with the Branch router:

```
Corp#sh ip ospf neighbor
Neighbor ID     Pri   State           Dead Time   Address         Interface
10.1.1.2          1   FULL/BDR        00:00:39    10.1.1.2        FastEthernet0/0
```

Okay, we've finally zeroed in on our problem—the Corp router can see the Internal router in area 0 but not the Branch router in area 1! What now?

First, let's review what we know so far about the Corp and Branch router. The data link is good, and we can use Ping successfully between the routers. This shouts out that we have a routing protocol issue, so we'll look further into the details of the OSPF configuration on each router. Let's run a `show ip protocols` on the Branch router:

```
Branch#sh ip protocols
Routing Protocol is "eigrp 20"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Default networks flagged in outgoing updates
  Default networks accepted from incoming updates
  EIGRP metric weight K1=1, K2=0, K3=1, K4=0, K5=0
  EIGRP maximum hopcount 100
  EIGRP maximum metric variance 1
  Redistributing: eigrp 20
  EIGRP NSF-aware route hold timer is 240s
  Automatic network summarization is not in effect
  Maximum path: 4
  Routing for Networks:
    10.0.0.0
    192.168.1.0
  Routing Information Sources:
    Gateway         Distance      Last Update
    (this router)         90      3d22h
    192.168.1.1           90      00:00:07
  Distance: internal 90 external 170


Routing Protocol is "ospf 2"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Router ID 192.168.1.2
  Number of areas in this router is 1. 1 normal 0 stub 0 nssa
  Maximum path: 4
  Routing for Networks:
    10.2.2.1 0.0.0.0 area 1
    192.168.1.2 0.0.0.0 area 1
 Reference bandwidth unit is 100 mbps
  Passive Interface(s):
    Serial0/0/0
  Routing Information Sources:
    Gateway         Distance      Last Update
    192.168.1.1          110      03:29:07
  Distance: (default is 110)
```

Do you see two routing protocols running on the Branch router? Both EIGRP and OSPF are running, but that's not necessarily our problem. The Corp router would need to be running EIGRP, and if so, we would have only EIGRP routes in our routing table because EIGRPs have the lower AD of 90 versus OSPF's AD of 110.

Let's check the routing table of the Branch router and see if the Corp router is also running EIGRP. This will be easy to determine if we discover EIGRP-injected routes in the table:

```
Branch#sh ip route
[output cut]
     10.0.0.0/24 is subnetted, 2 subnets
C       10.2.2.0 is directly connected, FastEthernet0/0
D       10.1.1.0 [90/2172416] via 192.168.1.1, 00:02:35, Serial0/0/0
     192.168.1.0/30 is subnetted, 1 subnets
C       192.168.1.0 is directly connected, Serial0/0/0
```

Okay—so yes, the Corp router is clearly running EIGRP. This is a leftover configuration from Chapter 5. All I need to do to fix this issue is to disable EIGRP on the Branch router. After that, we should see OSPF in the routing table:

```
Branch#config t
Branch(config)#no router eigrp 10
Branch(config)#do sh ip route
[output cut]
     10.0.0.0/24 is subnetted, 1 subnets
C       10.2.2.0 is directly connected, FastEthernet0/0
     192.168.1.0/30 is subnetted, 1 subnets
C       192.168.1.0 is directly connected, Serial0/0/0
```

That's not so good—I disabled the EIGRP protocol on the Branch router, but we still didn't receive OSPF updates! Let investigate further using the `show ip protocols` command on the Branch router:

```
Branch#sh ip protocols
Routing Protocol is "ospf 2"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Router ID 192.168.1.2
  Number of areas in this router is 1. 1 normal 0 stub 0 nssa
  Maximum path: 4
  Routing for Networks:
    10.2.2.1 0.0.0.0 area 1
    192.168.1.2 0.0.0.0 area 1
 Reference bandwidth unit is 100 mbps
  Passive Interface(s):
    Serial0/0/0
  Routing Information Sources:
    Gateway         Distance      Last Update
    192.168.1.1          110      03:34:19
  Distance: (default is 110)
```

Do you see the problem? There's no ACL, the networks are configured correctly, but see the passive interface for Serial0/0/0? That will definitely prevent an adjacency from happening between the Corp and Branch routers! Let's fix that:

```
Branach#show run
[output cut]
!
router ospf 2
 log-adjacency-changes
 passive-interface Serial0/0/0
 network 10.2.2.1 0.0.0.0 area 1
 network 192.168.1.2 0.0.0.0 area 1
!
[output cut]
Branch#config t
Branch(config)#router ospf 2
Branch(config-router)#no passive-interface serial 0/0/0
```

Let's see what our neighbor table and routing table look like now:

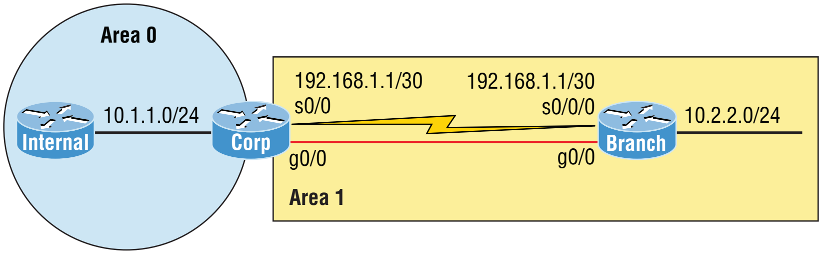
```
Branch#sh ip ospf neighbor
Neighbor ID     Pri   State           Dead Time   Address         Interface
192.168.1.1       0   FULL/  -        00:00:32    192.168.1.1     Serial0/0/0
```

```
Branch#sh ip route
     10.0.0.0/24 is subnetted, 2 subnets
C       10.2.2.0 is directly connected, FastEthernet0/0
O IA    10.1.1.0 [110/65] via 192.168.1.1, 00:01:21, Serial0/0/0
     192.168.1.0/30 is subnetted, 1 subnets
C       192.168.1.0 is directly connected, Serial0/0/0
```

Awesome—our little internetwork is finally happy! That was actually pretty fun and really not all that hard once you know what to look for.

But there's one more thing we need to cover before moving onto OSPFv3—load balancing with OSPF. To explore that, we'll use Figure 6.10, wherein I added another link between the Corp and Branch routers.

FIGURE 6.10 Our internetwork with dual links



First, it's clear that having a Gigabit Ethernet interface between our two routers is way better than any serial link we could possibly have, which means we want the routers to use the LAN link. We can either disconnect the serial link or use it as a backup link.

Let's start by looking at the routing table and seeing what OSPF found:

```
Corp#sh ip route ospf
     10.0.0.0/8 is variably subnetted, 3 subnets, 2 masks
O       10.2.2.0 [110/2] via 192.168.1.6, 00:00:13, GigabitEthernet0/1
```

Look at that! OSPF wisely went with the Gigabit Ethernet link because it has the lowest cost. Although it's possible you'll have to mess with the links to help OSPF choose the best paths, it's likely best to just leave it alone at this point.

But that wouldn't be very much fun, now would it? Instead, let's configure OSPF to fool it into thinking the links are equal so it will use both of them by setting the cost on the interfaces to the same value:

```
Corp#config t
Corp(config)#int g0/1
Corp(config-if)#ip ospf cost 10
Corp(config-if)#int s0/0/0
Corp(config-if)#ip ospf cost 10
```

Obviously you need to deploy this configuration on both sides of the link, and I've already configured the Branch router as well. Now that both sides are configured with the same cost, let's check out our routing table now:

```
Corp#sh ip route ospf
     10.0.0.0/8 is variably subnetted, 3 subnets, 2 masks
O       10.2.2.0 [110/11] via 192.168.1.2, 00:01:23, Serial0/0/0
                 [110/11] via 192.168.1.6, 00:01:23, GigabitEthernet0/1
```

I'm not saying you should configure a serial link and Gigabit Ethernet link as equal costs as I just demonstrated, but there are times when you need to adjust the cost for OSPF. If you don't have multiple links to any remote networks, you really don't need to worry about this, but with regard to the objectives, you absolutely must understand the cost, how it works, and how to set it so OSPF can choose a preferred path. And there's still one more thing about cost I want to cover with you.

It's possible to change the reference bandwidth of the router, but you need to make sure all the routers within the OSPF AS have the same reference bandwidth. The default reference bandwidth is 10^8, which is 100,000,000, or the equivalent of the bandwidth of Fast Ethernet, which is 100 Mbps, as demonstrated via `show ip ospf` and the `show ip protocols` command:

```
Routing for Networks:
    10.2.2.1 0.0.0.0 area 1
    192.168.1.2 0.0.0.0 area 1
Reference bandwidth unit is 100 mbps
```

This will basically make any interface running 100 Mbps or higher have a cost of 1. The default is 100, and if you change it to 1,000, it will increase the cost by a factor of 10. Again, if you do want to change this, you must make sure to configure the change on all routers in your AS! Here is how you would do that:

```
Corp(route)#router ospf 1
Corp(config-router)#auto-cost reference-bandwidth ?
  <1-4294967>  The reference bandwidth in terms of Mbits per second
```

Now, finally, let's get to the easy section of the chapter!

OSPFv3

The new version of OSPF continues the trend of routing protocols having a lot in common with their IPv4 versions. The foundation of OSPF remains the same—it's still a link-state routing protocol that divides an entire internetwork or autonomous system into areas, establishing a hierarchy.

In OSPF version 2, the router ID (RID) is determined by the highest IP addresses assigned to the router. And as you now know, the RID can be assigned. In version 3, nothing has really changed because you can still assign the RID, area ID, and link-state ID, which remain 32-bit values.

Adjacencies and next-hop attributes now use link-local addresses, but OSPFv3 still uses multi-cast traffic to send its updates and acknowledgements. It uses the addresses FF02::5 for OSPF routers and FF02::6 for OSPF-designated routers. These new addresses are the replacements for 224.0.0.5 and 224.0.0.6, respectively.

Other, less flexible IPv4 protocols don't give us the ability that OSPFv2 does to assign specific networks and interfaces into the OSPF process, but this is still configured under the router configuration process. And with OSPFv3, just as with the EIGRPv6 routing protocols we've talked about, the interfaces and therefore the networks attached to them are configured directly on the interface in interface configuration mode.

The configuration of OSPFv3 is going to look like this: First, optionally start by assigning the RID, but if you have IPv4 addresses assigned to your interface, you can let OSPF pick the RID just as we did with OSPFv2:

```
Router(config)#ipv6 router osfp 10
Router(config-rtr)#router-id 1.1.1.1
```

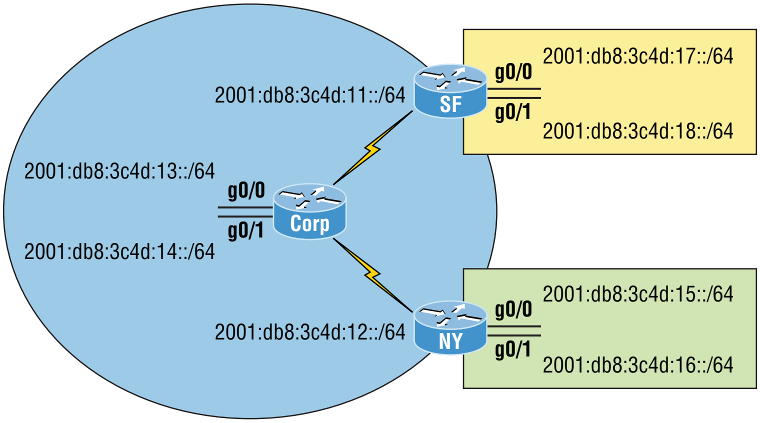
You get to perform some other configurations from router configuration mode, like summarization and redistribution, but again, we don't even need to configure OSPFv3 from this prompt if we configure it from the interface!

A simple interface configuration looks like this:

```
Router(config-if)#ipv6 ospf 10 area 0.0.0.0
```

So, if we just go to each interface and assign a process ID and area—poof, we're done! See? Easy! As the configuration shows, I configured the area as 0.0.0.0, which is the same thing as just typing `area 0`. We'll use Figure 6.11, which is the same network and IPv6-addressing we used in the EIGRPv6 section in Chapter 5.

FIGURE 6.11 Configuring OSPFv3



Okay, so all we have to do to enable OSPF on the internetwork is go to each interface that we want to run it on. Here's the Corp configuration:

```
Corp#config t
Corp(config)#int g0/0
Corp(config-if)#ipv6 ospf 1 area 0
Corp(config-if)#int g0/1
Corp(config-if)#ipv6 ospf 1 area 0
Corp(config-if)#int s0/0/0
Corp(config-if)#ipv6 ospf 1 area 0
Corp(config-if)#int s0/0/1
Corp(config-if)#ipv6 ospf 1 area 0
```

That wasn't so bad—much easier than it was with IPv4! To configure OSPFv3, you just need to establish the specific interfaces you'll be using! Let's configure the other two routers now:

```
SF#config t
SF(config)#int g0/0
SF(config-if)#ipv6 ospf 1 area 1
SF(config-if)#int g0/1
SF(config-if)#ipv6 ospf 1 area 1
SF(config-if)#int s0/0/0
SF(config-if)#ipv6 ospf 1 area 0
01:03:55: %OSPFv3-5-ADJCHG: Process 1, Nbr 192.168.1.5 on Serial0/0/0 from
LOADING to
FULL, Loading Done
```

Sweet—the SF has become adjacent to the Corp router! One interesting output line I want to point out is that the IPv4 RID is being used in the OSPFv3 adjacent change. I didn't set the RIDs manually because I knew I had interfaces with IPv4 addresses already on them, which the OSPF process would use for a RID.

Now let's configure the NY router:

```
NY(config)#int g0/0
NY(config-if)#ipv6 ospf 1 area 2
%OSPFv3-4-NORTRID:OSPFv3 process 1 could not pick a router-id,please configure
manually
NY(config-if)#ipv6 router ospf 1
NY(config-rtr)#router-id 1.1.1.1
NY(config-if)#int g0/0
NY(config-if)#ipv6 ospf 1 area 2
NY(config-if)#int g0/1
NY(config-if)#ipv6 ospf 1 area 2
NY(config-if)#int s0/0/0
NY(config-if)#ipv6 ospf 1 area 0
00:09:00: %OSPFv3-5-ADJCHG: Process 1, Nbr 192.168.1.5 on Serial0/0/0 from
LOADING to
FULL, Loading Done
```

Our adjacency popped up—this is great. But did you notice that I had to set the RID? That's because there wasn't an IPv4 32-bit address already on an interface for the router to use as the RID, so it was mandatory to set the RID manually!

Without even verifying our network, it appears it's up and running. Even so, it's always important to verify!

Verifying OSPFv3

I'll start as usual with the `show ipv6 route ospf` command:

```
Corp#sh ipv6 route ospf
OI  2001:DB8:3C4D:15::/64 [110/65]
     via FE80::201:C9FF:FED2:5E01, Serial0/0/1
OI  2001:DB8:3C4D:16::/64 [110/65]
     via FE80::201:C9FF:FED2:5E01, Serial0/0/1
O   2001:DB8:C34D:11::/64 [110/128]
     via FE80::2E0:F7FF:FE13:5E01, Serial0/0/0
OI  2001:DB8:C34D:17::/64 [110/65]
     via FE80::2E0:F7FF:FE13:5E01, Serial0/0/0
OI  2001:DB8:C34D:18::/64 [110/65]
     via FE80::2E0:F7FF:FE13:5E01, Serial0/0/0
Corp#
```

Perfect. I see all six subnets. Notice the `O` and `OI`? The `O` is intra-area and the `OI` is inter-area, meaning it's a route from a different area. You can't simply distinguish the area by looking at the routing table though. Plus, don't forget that the routers communicate with their neighbor via link-local addresses: via FE80::2E0:F7FF:FE13:5E01, Serial0/0/0, for example.

Let's take a look at the `show ipv6 protocols` command:

```
Corp#sh ipv6 protocols
IPv6 Routing Protocol is "connected"
IPv6 Routing Protocol is "static
IPv6 Routing Protocol is "ospf 1"
  Interfaces (Area 0)
    GigabitEthernet0/0
    GigabitEthernet0/1
    Serial0/0/0
    Serial0/0/1
```

This just tells us which interfaces are part of OSPF process 1, area 0. To configure OSPFv3, you absolutely have to know which interfaces are in use. `Sho ip int brief` can really help you if you're having a problem finding your active interfaces.

Let's take a look at the Gigabit Ethernet OSPFv3 active interface on the Corp router:

```
Corp#sh ipv6 ospf int g0/0
GigabitEthernet0/0 is up, line protocol is up
  Link Local Address FE80::2E0:F7FF:FE0A:3301 , Interface ID 1
  Area 0, Process ID 1, Instance ID 0, Router ID 192.168.1.5
  Network Type BROADCAST, Cost: 1
  Transmit Delay is 1 sec, State DR, Priority 1
  Designated Router (ID) 192.168.1.5, local address FE80::2E0:F7FF:FE0A:3301
  No backup designated router on this network
  Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
    Hello due in 00:00:09
  Index 1/1, flood queue length 0
  Next 0x0(0)/0x0(0)
  Last flood scan length is 1, maximum is 1
  Last flood scan time is 0 msec, maximum is 0 msec
  Neighbor Count is 0, Adjacent neighbor count is 0
  Suppress hello for 0 neighbor(s)
```

This is basically the same information we saw earlier in the verification and troubleshooting section. Let's take a look at the neighbor table on the Corp router via `show ipv6 ospf neighbor`:

```
Corp#sh ipv6 ospf neighbor
Neighbor ID     Pri   State           Dead Time   Interface ID    Interface
2.2.2.2           0   FULL/  -        00:00:36    4               Serial0/0/1
192.168.1.6       0   FULL/  -        00:00:39    4               Serial0/0/0
```

Okay, we can see our two neighbors, and there's also a slight difference in this version's command from OSPFv2. We still see the RID on the left and that we're also fully adjacent with both our neighbors—the dash is there because there are no elections on serial point-to-point links. But we don't see the neighbor's IPv6 address listed as we did with OSPFv2's IPv4 addreses, which were listed in the interface ID field.

There's one other command I want to finish with—the `show ipv6 ospf` command:

```
Corp#sh ipv6 ospf
 Routing Process "ospfv3 1" with ID 192.168.1.5
 SPF schedule delay 5 secs, Hold time between two SPFs 10 secs
 Minimum LSA interval 5 secs. Minimum LSA arrival 1 secs
 LSA group pacing timer 240 secs
 Interface flood pacing timer 33 msecs
 Retransmission pacing timer 66 msecs
```

```
Number of external LSA 0. Checksum Sum 0x000000
Number of areas in this router is 1. 1 normal 0 stub 0 nssa
Reference bandwidth unit is 100 mbps
   Area BACKBONE(0)
       Number of interfaces in this area is 4
       SPF algorithm executed 10 times
       Number of LSA 10. Checksum Sum 0x05aebb
       Number of DCbitless LSA 0
       Number of indication LSA 0
       Number of DoNotAge LSA 0
       Flood list length 0
```

This shows the process ID and RID, our reference bandwidth for this interface, and how many interfaces we have in each area, which in our example is only area 0.

Holy output! Now that's what I call a fun chapter. The best thing you can do to get a solid grasp of OSPF and OSPv3 multi-area networks is to gather up some routers and spend some quality time with them, practicing everything we've covered!

Summary

In this chapter, you learned about the scalability constraints of a single-area OSPF network, and you were introduced to the concept of multi-area OSPF as a solution to these scalability limitations.

You're now able to identify the different categories of routers used in multi-area configurations, including the backbone router, internal router, area border router, and autonomous system boundary router.

I detailed the function of different OSPF Link-State Advertisements (LSAs) and you discovered how these LSAs can be minimized through the effective implementation of specific OSPF area types. I discussed the Hello protocols and the different neighbor states experienced when an adjacency is taking place.

Verification and troubleshooting are very large parts of the objectives, and I covered everything you need to know in order to verify and troubleshoot OSPFv2 and meet those requirements.

Finally, we ended the chapter with the easiest part: configuring and verifying OSPFv3.

Exam Essentials

Know the scalability issues multi-area OSPF addresses. The primary problems in single-area OSPF networks are the large size of the topology and routing tables, as well as, the excessive computation of the SPF algorithm due to the large number of link-state updates that occur in this single area.

Know the different types of OSPF routers. Backbone routers have at least one interface in area 0. Area border routers (ABRs) belong to two or more OSPF areas simultaneously. Internal routers have all of their interfaces within the same area. Autonomous system boundary routers (ASBRs) have at least one interface connected to an external network.

Know the different types of LSA packets. There are seven different types of LSA packets that Cisco uses, but here are the ones you need to remember: Type 1 LSAs (router link advertisements), Type 2 LSAs (network link advertisements), Type 3 and 4 LSAs (summary LSAs), and Type 5 LSAs (AS external link advertisements). Know how each functions.

Be able to monitor multi-area OSPF. There are a number of commands that provide information useful in a multi-area OSPF environment: `show ip route ospf`, `show ip ospf neighbor`, `show ip ospf`, and `show ip ospf database`. It's important to understand what each provides.

Be able to troubleshoot OSPF networks. It's important that you can work your way through the troubleshooting scenario that I presented in this chapter. Be able to look for neighbor adjacencies, and if they are not there, look for ACLs set on the routing protocol, passive interfaces, and wrong network statements.

Understand how to configure OSPFv3. OSPFv3 uses the same basic mechanisms that OSPFv2 uses, but OSPFv3 is more easily configured by placing the configuring OSPFv3 on a per-interface basis with the `ipv6 ospf` *`process-ID`* `area` *`area`*.

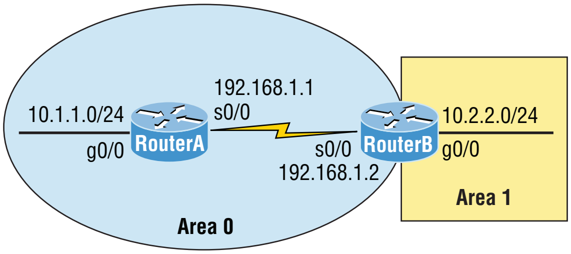
Written Lab 6

The answers to this lab can be found in Appendix A, "Answers to Written Labs."

1. What type of LSAs are sent by an ASBR?
2. What state would a router adjacency be in after the INIT state has finished?
3. What LSA types are sent by ABR toward the area external to the one in which they were generated?
4. When would you see an adjacency show this: `FULL/-`?
5. True/False: OSPFv3 is configured per area, per interface.
6. Which OSPF state uses DBD packets and LSRs?
7. Which LSA type is referred to as a router link advertisement (RLA)?
8. What is the command to configure OSPFv3 on an interface with process ID 1 into area 0?
9. What must match exactly between two routers to form an adjacency when using OSPFv3?
10. How can you see all the routing protocols configured and running on your router from user mode?

Hands-on Labs

In this section, you will use the following network and add OSPF and OSPFv3 routing.



The first lab requires you to configure two routers with OSPF and then verify the configuration. In the second, you will be asked to enable OSPFv3 routing on the same network. Note that the labs in this chapter were written to be used with real equipment—real cheap equipment, that is. As with Chapter 5 on EIGRP, I wrote these labs with the cheapest, oldest routers I had lying around so you can see that you don't need expensive gear to get through some of the hardest labs in this book. However, you can use the free LammleSim IOS version simulator or Cisco's Packet Tracer to run through these labs.

The labs in this chapter are as follows:

Lab 6.1: Configuring and Verifying Multi-Area OSPF

Lab 6.2: Configuring and Verifying OSPFv3

Hands-on Lab 6.1: Configuring and Verifying OSPF Multi-Area

In this lab, you'll configure and verify multi-area OSPF:

1. Implement OSPFv2 on RouterA based on the information in the diagram.

```
RouterA#conf t
RouterA(config)#router ospf 10
RouterA(config-router)#network 10.0.0.0 0.255.255.255 area 0
RouterA(config-router)#network 192.168.1.0 0.0.0.255 area 0
```

2. Implement OSPF on RouterB based on the diagram.

```
RouterB#conf t
RouterB(config)#router ospf 1
RouterB(config-router)#network 192.168.1.2 0.0.0.0 area 0
RouterB(config-router)#network 10.2.2.0 0.0.0.255 area 1
```

3. Display all the LSAs received on RouterA.

```
RouterA#sh ip ospf database

            OSPF Router with ID (192.168.1.1) (Process ID 10)

                Router Link States (Area 0)

Link ID         ADV Router      Age         Seq#       Checksum Link count
10.1.1.2        10.1.1.2        380         0x80000035 0x0012AB 1
192.168.1.1     192.168.1.1     13          0x8000000A 0x00729F 3
192.168.1.2     192.168.1.2     10          0x80000002 0x0090F9 2

                Net Link States (Area 0)
```

```
Link ID         ADV Router      Age         Seq#       Checksum
10.1.1.2        10.1.1.2        381         0x80000001 0x003371


                Summary Net Link States (Area 0)


Link ID         ADV Router      Age         Seq#       Checksum
10.2.2.0        192.168.1.2     8           0x80000001 0x00C3FD
RouterA#
```

4. Display the routing table for RouterA.

```
RouterA#sh ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia -IS-IS inter area,* - candidate default,U - per-user static route
       o - ODR, P - periodic downloaded static route


Gateway of last resort is not set


     10.0.0.0/24 is subnetted, 2 subnets
O IA    10.2.2.0 [110/101] via 192.168.1.2, 00:00:29, Serial0/0
C       10.1.1.0 is directly connected, FastEthernet0/0
     192.168.1.0/30 is subnetted, 1 subnets
C       192.168.1.0 is directly connected, Serial0/0
RouterA#
```

5. Display the neighbor table for RouterA.

```
RouterA#sh ip ospf neighbor


Neighbor ID     Pri   State        Dead Time   Address        Interface
192.168.1.2       0   FULL/  -     00:00:35    192.168.1.2    Serial0/0
10.1.1.2          1   FULL/DR      00:00:34    10.1.1.2       FastEthernet0/0
RouterA#
```

6. Use the show `ip ospf` command on RouterB to see that it is an ABR.

```
RouterB#sh ip ospf
 Routing Process "ospf 1" with ID 192.168.1.2
 Start time: 1w4d, Time elapsed: 00:07:04.100
 Supports only single TOS(TOS0) routes
 Supports opaque LSA
 Supports Link-local Signaling (LLS)
 Supports area transit capability
 It is an area border router
 Router is not originating router-LSAs with maximum metric
 Initial SPF schedule delay 5000 msecs
 Minimum hold time between two consecutive SPFs 10000 msecs
 Maximum wait time between two consecutive SPFs 10000 msecs
 Incremental-SPF disabled
 Minimum LSA interval 5 secs
 Minimum LSA arrival 1000 msecs
 LSA group pacing timer 240 secs
 Interface flood pacing timer 33 msecs
 Retransmission pacing timer 66 msecs
 Number of external LSA 0. Checksum Sum 0x000000
 Number of opaque AS LSA 0. Checksum Sum 0x000000
 Number of DCbitless external and opaque AS LSA 0
 Number of DoNotAge external and opaque AS LSA 0
 Number of areas in this router is 2. 2 normal 0 stub 0 nssa
 Number of areas transit capable is 0
 External flood list length 0
    Area BACKBONE(0)
        Number of interfaces in this area is 1
        Area has no authentication
        SPF algorithm last executed 00:06:44.492 ago
        SPF algorithm executed 3 times
        Area ranges are
        Number of LSA 5. Checksum Sum 0x020DB1
        Number of opaque link LSA 0. Checksum Sum 0x000000
        Number of DCbitless LSA 0
        Number of indication LSA 0
        Number of DoNotAge LSA 0
        Flood list length 0
    Area 1
        Number of interfaces in this area is 1
```

```
Area has no authentication
SPF algorithm last executed 00:06:45.640 ago
SPF algorithm executed 2 times
Area ranges are
Number of LSA 3. Checksum Sum 0x00F204
Number of opaque link LSA 0. Checksum Sum 0x000000
Number of DCbitless LSA 0
Number of indication LSA 0
Number of DoNotAge LSA 0
Flood list length 0
```

Hands-on Lab 6.2: Configuring and Verifying OSPFv3

In this lab, you will configure and verify OSPFv3:

1. Implement OSPFv3 on RouterA. Since the routers have IPv4 addresses, we don't need to set the RID of the router.

   ```
   RouterA#config t
   RouterA(config)#int g0/0
   RouterA(config-if)#ipv6 ospf 1 area 0
   RouterA(config-if)#int s0/0
   RouterA(config-if)#ipv6 ospf 1 area 0
   ```

   That's all there is to it! Nice.

2. Implement OSPFv3 on RouterB.

   ```
   RouterB#config t
   RouterB(config)#int s0/0/0
   RouterB(config-if)#ipv6 ospf 1 area 0
   RouterB(config-if)#int f0/0
   RouterB(config-if)#ipv6 ospf 1 area 1
   ```

   Again, that's all there is to it!

3. Display the routing table for RouterA.

   ```
   RouterA#sh ipv6 route ospf
   IPv6 Routing Table - 11 entries
   Codes: C - Connected, L - Local, S - Static, R - RIP, B - BGP
          U - Per-user Static route
          I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary
          O - OSPF intra, OI - OSPF inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2
   ```

```
       ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2
       D - EIGRP, EX - EIGRP external
OI  2001:DB8:3C4D:15::/64 [110/65]
     via FE80::21A:2FFF:FEE7:4398, Serial0/0
RouterA#
```

Notice that the one route OSPFv3 found is an inter-area route, meaning the network is another area.

4. Display the neighbor table for RouterA.

```
RouterA#sh ipv6 ospf neighbor


Neighbor ID     Pri   State           Dead Time   Interface ID    Interface
192.168.1.2       1   FULL/  -        00:00:32    6               Serial0/0
RouterA#
```

5. Display the `show ipv6 ospf` command on RouterB.

```
RouterB#sh ipv6 ospf
 Routing Process "ospfv3 1" with ID 192.168.1.2
 It is an area border router
 SPF schedule delay 5 secs, Hold time between two SPFs 10 secs
 Minimum LSA interval 5 secs. Minimum LSA arrival 1 secs
 LSA group pacing timer 240 secs
 Interface flood pacing timer 33 msecs
 Retransmission pacing timer 66 msecs
 Number of external LSA 0. Checksum Sum 0x000000
 Number of areas in this router is 2. 2 normal 0 stub 0 nssa
 Reference bandwidth unit is 100 mbps
    Area BACKBONE(0)
        Number of interfaces in this area is 1
        SPF algorithm executed 3 times
        Number of LSA 7. Checksum Sum 0x041C1B
        Number of DCbitless LSA 0
        Number of indication LSA 0
        Number of DoNotAge LSA 0
        Flood list length 0
    Area 1
        Number of interfaces in this area is 1
        SPF algorithm executed 2 times
        Number of LSA 5. Checksum Sum 0x02C608
        Number of DCbitless LSA 0
        Number of indication LSA 0
        Number of DoNotAge LSA 0
        Flood list length 0

RouterB#
```

Review Questions

NOTE The following questions are designed to test your understanding of this chapter's material. For more information on how to get additional questions, please see this book's introduction.

The answers to these questions can be found in Appendix B, "Answers to Chapter Review Questions."

1. Which of the following are scalability issues with single-area OSPF networks? (Choose all that apply.)
   - **A.** Size of the routing table
   - **B.** Size of the OSPF database
   - **C.** Maximum hop-count limitation
   - **D.** Recalculation of the OSPF database
2. Which of the following describes a router that connects to an external routing process (e.g., EIGRP)?
   - **A.** ABR
   - **B.** ASBR
   - **C.** Type 2 LSA
   - **D.** Stub router
3. Which of the following must match in order for an adjacency to occur between routers? (Choose three.)
   - **A.** Process ID
   - **B.** Hello and dead timers
   - **C.** Link cost
   - **D.** Area
   - **E.** IP address/subnet mask
4. Which OSPF state do two routers forming an adjacency appear as in the `show ip ospf neighbor` output after adding neighbors into the table and exchanging hello information?
   - **A.** `ATTEMPT`
   - **B.** `INIT`
   - **C.** `2WAY`
   - **D.** `EXSTART`
   - **E.** `FULL`

5. You need to set up a preferred link that OSPF will use to route information to a remote network. Which command will allow you to set the interface link as preferred over another?

   A. `ip ospf preferred 10`

   B. `ip ospf priority 10`

   C. `ospf bandwidth 10`

   D. `ip ospf cost 10`

6. When would a router's neighbor table show the `FULL/DR` state?

   A. After the first Hello packets are received by a neighbor

   B. When all information is synchronized among adjacent neighbors

   C. When the router's neighbor table is too full of information and is discarding neighbor information

   D. After the `EXSTART` state

7. Which is/are true regarding OSPFv3? (Choose all that apply.)

   A. You must add network statements under the OSPF process.

   B. There are no network statements in OSPFv3 configurations.

   C. OSPFv3 uses a 128-bit RID.

   D. If you have IPv4 configured on the router, it is not mandatory that you configure the RID.

   E. If you don't have IPv4 configured on the router, it is mandatory that you configure the RID.

   F. OSPFv3 doesn't use LSAs like OSPFv2 does.

8. What is the default cost of a Fast Ethernet interface configured with OSPF?

   A. 1

   B. 10

   C. 100

   D. 1,000

9. Which type of LSA is generated by DRs and referred to as a network link advertisement (NLA)?

   A. Type 1

   B. Type 2

   C. Type 3

   D. Type 4

   E. Type 5

10. Which type of LSA is generated by ABRs and refers to a summary link advertisement (SLA)?

A. Type 1

B. Type 2

C. Type 3

D. Type 4

E. Type 5

11. Which command will show all the LSAs known by a router?

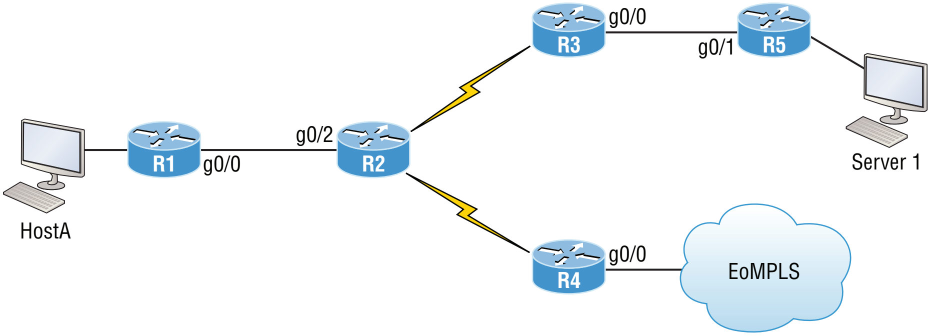
A. `show ip ospf`

B. `show ip ospf neighbor`

C. `show ip ospf interface`

D. `show ip ospf database`

12. Using the following illustration, what is the cost from R1's routing table to reach the network with Server 1? Each Gigabit Ethernet link has a cost of 4, and each serial link has a cost of 15.



A. 100

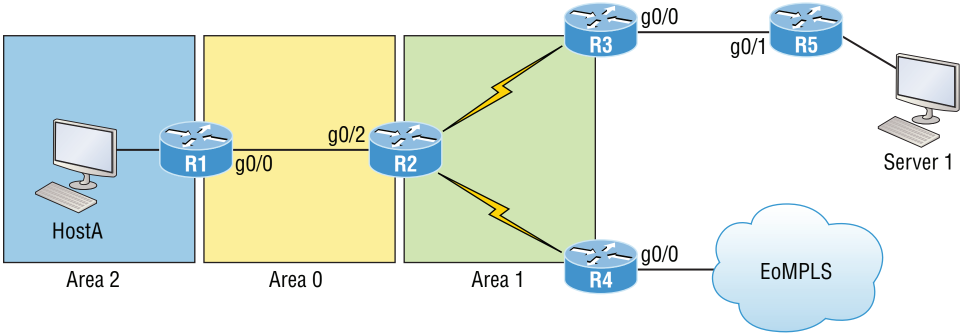
B. 23

C. 64

D. 19

E. 27

13. Using the following illustration, which of the following are true? (Choose all that apply.)



A. R1 is an internal router.

B. R3 would see the networks connected to the R1 router as an inter-area route.

C. R2 is an ASBR.

D. R3 and R4 would receive information from R2 about the backbone area, and the same LSA information would be in both LSDBs.

E. R4 is an ABR.

14. Which of the following could cause two routers to not form an adjacency? (Choose all that apply.)

A. They are configured in different areas.

B. Each router sees the directly connected link as different costs.

C. Two different Process ID's configured.

D. ACL is configured on the routing protocol.

E. IP address/mask.

F. Passive interface is configured.

15. Which of the following IOS commands shows the state of an adjacency with directly connected routers?

A. `debug ospf events`

B. `show ip ospf border-routers`

C. `show ip ospf neighbor`

D. `show ip ospf database`

16. What command will show you the DR and DBR address of the area you are connected to directly with an interface?
    - A. `show interface s0/0/0`
    - B. `show interface fa0/0`
    - C. `show ip ospf interface s0/0/0`
    - D. `show ip ospf interface fa0/0`

17. Which of the following could be causing a problem with the Corp router not forming an adjacency with its neighbor router? (Choose all that apply.)

```
Corp#sh ip protocols
Routing Protocol is "ospf 1"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is 10
  Router ID 1.1.1.1
  Number of areas in this router is 3. 3 normal 0 stub 0 nssa
  Maximum path: 4
  Routing for Networks:
    10.10.0.0 0.0.255.255 area 0
    172.16.10.0 0.0.0.3 area 1
    172.16.10.4 0.0.0.3 area 2
 Reference bandwidth unit is 100 mbps
  Passive Interface(s):
    Serial0/0/0
 Routing Information Sources:
    Gateway         Distance      Last Update
    1.1.1.1              110      00:17:42
    172.16.10.2          110      00:17:42
    172.16.10.6          110      00:17:42
  Distance: (default is 110)
```

- A. The routers are configured with the wrong network statements.
- B. They have different maximum paths configured.
- C. There is a passive interface configured.
- D. There is an ACL set stopping Hellos.
- E. The costs of the links between the routers are configured differently.
- F. They are in different areas.

18. Which of the following is/are true? (Choose all that apply.)

A. The reference bandwidth for OSPF and OSPFv3 is 1.

B. The reference bandwidth for OSPF and OSPFv3 is 100.

C. You change the reference bandwidth from global config with the command auto-cost reference bandwidth number.

D. You change the reference bandwidth under the OSPF router process with the command auto-cost reference bandwidth number.

E. Only one router needs to set the reference bandwidth if it is changed from its default.

F. All routers in a single area must set the reference bandwidth if it is changed from its default.

G. All routers in the AS must set the reference bandwidth if it is changed from its default.

19. Which of the following interfaces would have the same default cost? (Choose two.)

A. Fast Ethernet

B. Ethernet

C. Serial

D. Gigabit Ethernet

20. What is the default cost of a serial interface with OSPF?

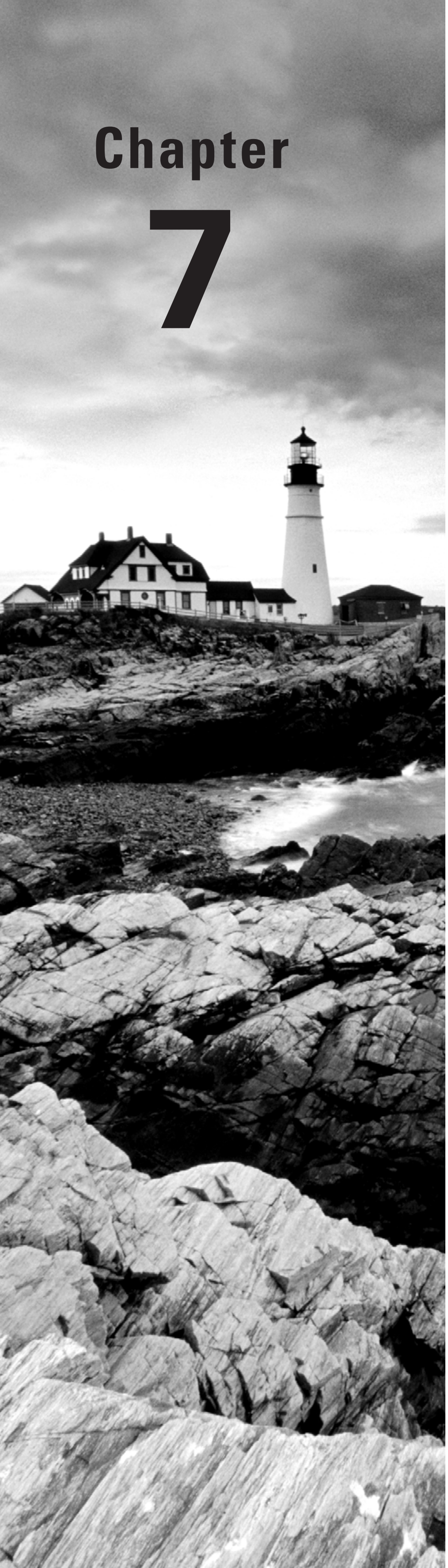
A. 1

B. 10

C. 32

D. 64

E. 100



Chapter 7

Wide Area Networks

THE FOLLOWING ICND2 EXAM TOPICS ARE COVERED IN THIS CHAPTER:

✓ **IP Routing Technologies**

- Configure and verify operation status of a Serial interface

✓ **WAN Technologies**

- Identify different WAN Technologies
  - Metro Ethernet
  - VSAT
  - Cellular 3G/4G
  - MPLS
  - T1/E1
  - ISDN
  - DSL
  - Frame relay
  - Cable
  - VPN
- Configure and verify a basic WAN serial connection
- Configure and verify a PPP connection between Cisco routers
- Configure and verify Frame Relay on Cisco routers
- Implement and troubleshoot PPPoE

✓ **Troubleshooting**

- Troubleshoot and resolve WAN implementation issues
  - Serial interfaces
  - PPP
  - Frame Relay

The Cisco IOS supports a ton of different wide area network (WAN) protocols that help you extend your local LANs to other LANs at remote sites. And I don't think I have to tell you how essential information exchange between disparate sites is these days—it's absolutely vital! But even so, it wouldn't exactly be cost effective or efficient to install your own cable and connect all of your company's remote locations yourself, would it? A much better way to get this done is to just lease the existing installations that service providers already have in place.

This is exactly why I'm going to devote most of this chapter to covering the various types of connections, technologies, and devices used in today's WANs. We'll also delve into how to implement and configure High-Level Data-Link Control (HDLC), Point-to-Point Protocol (PPP), and Frame Relay. I'll describe Point-to-Point Protocol over Ethernet (PPPoE), cable, digital subscriber line (DSL), MultiProtocol Label Switching (MPLS), Metro Ethernet, plus last mile and long-range WAN technologies. I'll also introduce you to WAN security concepts, tunneling, and virtual private network basics.

Just so you know, I'm not going to cover each and every type of Cisco WAN support here because the focus of this book is to equip you with everything you need to successfully meet the Cisco exam objectives. This chapter's emphasis will be on cable, DSL, HDLC, PPP, PPPoE, Metro Ethernet, MPLS, and Frame Relay, but I'll wrap it up by giving you a solid grounding in virtual private networks (VPNs) and how to create a tunnel using GRE (Generic Routing Encapsulation).

NOTE

To find up-to-the-minute updates for this chapter, please see `www.lammle.com/forum` or the book's web page at `www.sybex.com`.

Introduction to Wide Area Networks

Let's begin exploring WAN basics by asking, What's the difference between a *wide area network (WAN)* and a local area network (LAN)? Clearly there's the distance factor, but modern wireless LANs can cover some serious turf, so there's more to it than that. What about bandwidth? Here again, some really big pipes can be had for a price in many places, so that's not it either. What's the answer we're looking for?

A major distinction between a WAN and a LAN is that while you generally own a LAN infrastructure, you usually lease a WAN infrastructure from a service provider. And to be honest, modern technologies even blur this characteristic somewhat, but it still fits neatly into the context of Cisco's exam objectives!

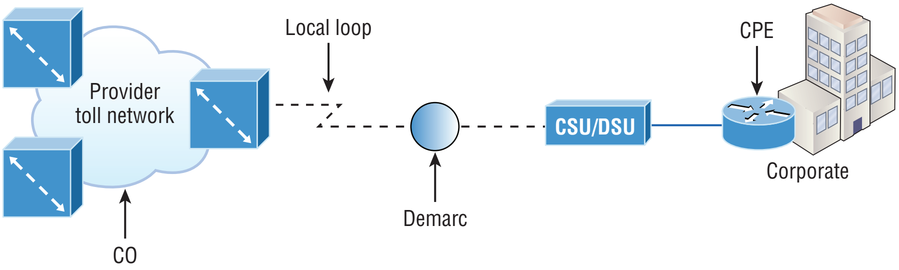
I've already talked about the Data Link that you usually own when we covered Ethernet, so I'm going to focus on the type you usually don't own—the kind you typically lease from a service provider.

And as usual, the first key to understanding WAN technologies is to be familiar with the different WAN terms and connection types commonly used by service providers to join your LAN networks together. We'll begin covering these topics now.

Defining WAN Terms

Before you run out and order a WAN service type from a provider, you really need to understand the following terms that service providers typically use. Take a look at these in Figure 7.1:

FIGURE 7.1 WAN terms



Customer premises equipment (CPE) *Customer premises equipment (CPE)* is equipment that's typically owned by the subscriber and located on the subscriber's premises.

Demarcation point The *demarcation point* (demarc for short) is the precise spot where the service provider's responsibility ends and the CPE begins. It's generally a device in a telecommunications closet owned and installed by the telecommunications company (telco). It's your responsibility to cable (extended demarc) from this box to the CPE, which is usually a connection to a CSU/DSU, although more recently we see the provider giving us an Ethernet connection. Nice!

Local loop The *local loop* connects the demarc to the closest switching office, referred to as the central office.

Central office (CO) This point connects the customer's network to the provider's switching network. Make a mental note that a *central office (CO)* is sometimes also referred to as a *point of presence (POP)*.

Toll network The *toll network* is a trunk line inside a WAN provider's network. This network is a collection of switches and facilities owned by the Internet service provider (ISP).

Definitely familiarize yourself with these terms, what they represent, and where they're located, as shown in Figure 7.1, because they're key to understanding WAN technologies.

WAN Connection Bandwidth

Next, I want you to know these basic but very important bandwidth terms used when referring to WAN connections:

Digital Signal 0 (DS0) This is the basic digital signaling rate of 64 Kbps, equivalent to one channel. Europe uses the E0 and Japan uses the J0 to reference the same channel speed. Typical to T-carrier transmission, this is the generic term used by several multiplexed digital carrier systems and is also the smallest-capacity digital circuit. 1 DS0 = 1 voice/data line.

T1 Also referred to as a DS1, a T1 comprises 24 DS0 circuits bundled together for a total bandwidth of 1.544 Mbps.

E1 This is the European equivalent of a T1 and comprises 30 DS0 circuits bundled together for a bandwidth of 2.048 Mbps.

T3 Referred to as a DS3, a T3 comprises 28 DS1s bundled together, or 672 DS0s, for a bandwidth of 44.736 Mbps.

OC-3 Optical Carrier (OC) 3 uses fiber and is made up of three DS3s bundled together. It's made up of 2,016 DS0s and avails a total bandwidth of 155.52 Mbps.

OC-12 Optical Carrier 12 is made up of four OC-3s bundled together and contains 8,064 DS0s for a total bandwidth of 622.08 Mbps.

OC-48 Optical Carrier 48 is made up of four OC-12s bundled together and contains 32,256 DS0s for a total bandwidth of 2488.32 Mbps.

WAN Connection Types

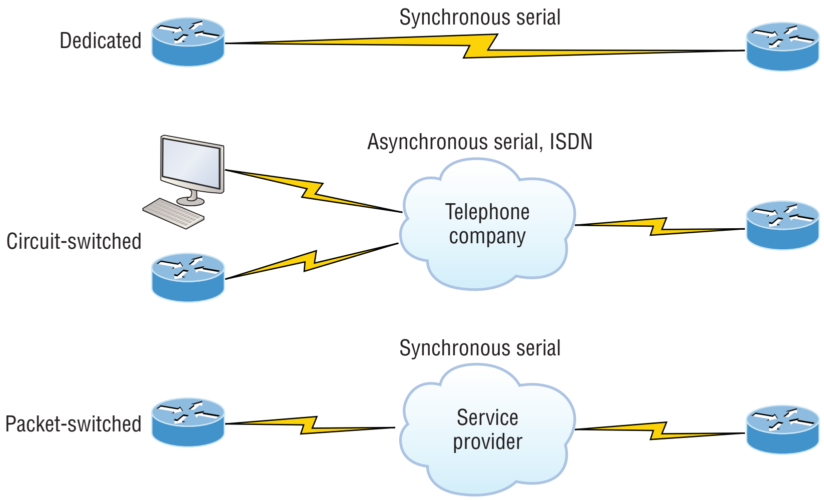
You're probably aware that a WAN can use a number of different connection types available on the market today. Figure 7.2 shows the different WAN connection types that can be used to connect your LANs (made up of data terminal equipment, or DTE) together over the data communication equipment (DCE) network.

Let me explain the different WAN connection types in detail now:

Dedicated (leased lines) These are usually referred to as a *point-to-point* or dedicated connections. A *leased line* is a pre-established WAN communications path that goes from the CPE through the DCE switch, then over to the CPE of the remote site. The CPE enables DTE networks to communicate at any time with no cumbersome setup procedures to muddle through before transmitting data. When you've got plenty of cash, this is definitely the way to go because it uses synchronous serial lines up to 45 Mbps. HDLC and PPP encapsulations are frequently used on leased lines, and I'll go over these with you soon.

Circuit switching When you hear the term *circuit switching*, think phone call. The big advantage is cost; most plain old telephone service (POTS) and ISDN dial-up connections are not flat rate, which is their advantage over dedicated lines. No data can transfer before an end-to-end connection is established. Circuit switching uses dial-up modems or ISDN and is used for low-bandwidth data transfers. Okay, I know what you're thinking: "Modems? Did he say modems? Aren't those found only in museums now?" After all, with all the wireless technologies available, who would use a modem these days? Well, some people do have ISDN; it's still viable and there are a few who still use a modem now and then. And circuit switching can be used in some of the newer WAN technologies as well.

FIGURE 7.2 WAN connection types



Packet switching This is a WAN switching method that allows you to share bandwidth with other companies to save money, just like a super old party line, where homes shared the same phone number and line to save money. *Packet switching* can be thought of as a network that's designed to look like a leased line yet it charges you more, like circuit switching does. As usual, you get what you pay for, and there's definitely a serious downside to this technology. If you need to transfer data constantly, well, just forget about this option and get a leased line instead! Packet switching will only really work for you if your data transfers are bursty, not continuous; think of a highway, where you can only go as fast as the traffic—packet switching is the same thing. Frame Relay and X.25 are packet-switching technologies with speeds that can range from 56 Kbps up to T3 (45 Mbps).

NOTE MultiProtocol Label Switching (MPLS) uses a combination of both circuit switching and packet switching, but it's not within the scope of topics this book covers. Still, because I would highly recommend checking into MPLS after you pass your CCNA R/S exam, I'll talk about it briefly in a minute.

WAN Support

Cisco supports many layer 2 WAN encapsulations on its serial interfaces, including HDLC, PPP, and Frame Relay, which map to the Cisco exam objectives. You can view them via the `encapsulation ?` command from any serial interface, but understand that the output you'll get can vary based upon the specific IOS version you're running:

```
Corp#config t
Corp(config)#int s0/0/0
Corp(config-if)#encapsulation ?
  atm-dxi      ATM-DXI encapsulation
  frame-relay  Frame Relay networks
  hdlc         Serial HDLC synchronous
  lapb         LAPB (X.25 Level 2)
  ppp          Point-to-Point protocol
  smds         Switched Megabit Data Service (SMDS)
  x25          X.25
```

I also want to point out that if I had other types of interfaces on my router, I would have a different set of encapsulation options. And never forget that you can't configure an Ethernet encapsulation on a serial interface or vice versa!

Next, I'm going to define the most prominently known WAN protocols used in the latest Cisco exam objectives: Frame Relay, ISDN, HDLC, PPP, PPPoE, cable, DSL, MPLS, ATM, 3G/4G, VSAT, and Metro Ethernet. Just so you know, the only WAN protocols you'll usually find configured on a serial interface are HDLC, PPP, and Frame Relay, but who said you're stuck with using only serial interfaces for wide area connections? Actually, we're beginning to see fewer and fewer serial connections because they're not as scalable or cost effective as a Fast Ethernet connection to your ISP.

Frame Relay A packet-switched technology that made its debut in the early 1990s, *Frame Relay* is a high-performance Data Link and Physical layer specification. It's pretty much a successor to X.25, except that much of the technology in X.25 used to compensate for physical errors like noisy lines has been eliminated. An upside to Frame Relay is that it can be more cost effective than point-to-point links, plus it typically runs at speeds of 64 Kbps up to 45 Mbps (T3). Another Frame Relay benefit is that it provides features for dynamic bandwidth allocation and congestion control.

ISDN *Integrated Services Digital Network (ISDN)* is a set of digital services that transmit voice and data over existing phone lines. ISDN offers a cost-effective solution for remote users who need a higher-speed connection than analog POTS dial-up links can give them, and it's also a good choice to use as a backup link for other types of links, such as Frame Relay or T1 connections.

HDLC *High-Level Data-Link Control (HDLC)* was derived from Synchronous Data Link Control (SDLC), which was created by IBM as a Data Link connection protocol. HDLC works at the Data Link layer and creates very little overhead compared to Link Access Procedure, Balanced (LAPB).

Generic HDLC wasn't intended to encapsulate multiple Network layer protocols across the same link—the HDLC header doesn't contain any identification about the type of protocol being carried inside the HDLC encapsulation. Because of this, each vendor that uses HDLC has its own way of identifying the Network layer protocol, meaning each vendor's HDLC is proprietary with regard to its specific equipment.

PPP *Point-to-Point Protocol (PPP)* is a pretty famous, industry-standard protocol. Because all multiprotocol versions of HDLC are proprietary, PPP can be used to create point-to-point links between different vendors' equipment. It uses a Network Control Protocol field in the Data Link header to identify the Network layer protocol being carried and allows authentication and multilink connections to be run over asynchronous and synchronous links.

PPPoE *Point-to-Point Protocol over Ethernet* encapsulates PPP frames in Ethernet frames and is usually used in conjunction with xDSL services. It gives you a lot of the familiar PPP features like authentication, encryption, and compression, but there's a downside—it has a lower maximum transmission unit (MTU) than standard Ethernet does. If your firewall isn't solidly configured, this little factor can really give you some grief!

Still somewhat popular in the United States, PPPoE's main feature is that it adds a direct connection to Ethernet interfaces while also providing DSL support. It's often used by many hosts on a shared Ethernet interface for opening PPP sessions to various destinations via at least one bridging modem.

Cable In a modern *hybrid fiber-coaxial (HFC)* network, typically 500 to 2,000 active data subscribers are connected to a certain cable network segment, all sharing the upstream and downstream bandwidth. HFC is a telecommunications industry term for a network that incorporates both optical fiber and coaxial cables to create a broadband network. The actual bandwidth for Internet service over a cable TV (CATV) line can be up to about 27 Mbps on the download path to the subscriber, with about 2.5 Mbps of bandwidth on the upload path. Typically users get an access speed from 256 Kbps to 6 Mbps. This data rate varies greatly throughout the United States and can be much, much higher today.

DSL Digital subscriber line is a technology used by traditional telephone companies to deliver advanced services such as high-speed data and sometimes video over twisted-pair copper telephone wires. It typically has lower data-carrying capacity than HFC networks, and data speeds can be limited in range by line lengths and quality. Digital subscriber line is not a complete end-to-end solution but rather a Physical layer transmission technology like dial-up, cable, or wireless. DSL connections are deployed in the last mile of a local telephone network—the local loop. The connection is set up between a pair of DSL modems on either end of a copper wire located between the customer premises equipment (CPE) and the Digital Subscriber Line Access Multiplexer (DSLAM). A DSLAM is the device located at the provider's central office (CO) and concentrates connections from multiple DSL subscribers.

MPLS *MultiProtocol Label Switching (MPLS)* is a data-carrying mechanism that emulates some properties of a circuit-switched network over a packet-switched network. MPLS is a switching mechanism that imposes labels (numbers) to packets and then uses them to forward packets. The labels are assigned on the edge of the MPLS network, and forwarding inside the MPLS network is carried out solely based on the labels. The labels usually correspond to a path to layer 3 destination addresses, which is on par with IP destination-based routing. MPLS was designed to support the forwarding of protocols other than TCP/IP. Because of this, label switching within the network is achieved the same way irrespective of the layer 3 protocol. In larger networks, the result of MPLS labeling is that only the edge routers perform a routing lookup. All the core routers forward packets based on the labels, which makes forwarding the packets through the service provider network faster. This is a big reason most companies have replaced their Frame Relay networks with MPLS service today. Last, you can use Ethernet with MPLS to connect a WAN, and this is called Ethernet over MPLS, or EoMPLS.

ATM Asynchronous Transfer Mode (ATM) was created for time-sensitive traffic, providing simultaneous transmission of voice, video, and data. ATM uses cells that are a fixed 53-bytes long instead of packets. It also can use isochronous clocking (external clocking) to help the data move faster. Typically, if you're running Frame Relay today, you will be running Frame Relay over ATM.

Cellular 3G/4G Having a wireless hot spot in your pocket is pretty normal these days. If you have a pretty current cellular phone, then you can probably can gain access through your phone to the Internet. You can even get a 3G/4G card for an ISR router that's useful for a small remote office that's in the coverage area.

VSAT Very Small Aperture Terminal (VSAT) can be used if you have many locations geographically spread out in a large area. VSAT uses a two-way satellite ground station with dishes available through many companies like Dish Network or Hughes and connects to satellites in geosynchronous orbit. A good example of where VSATs are a useful, cost-effective solution would be companies that use satellite communications to VSATs, like gasoline stations that have hundreds or thousands of locations spread out over the entire country. How could you connect them otherwise? Using leased lines would be cost prohibitive and dial-ups would be way too slow and hard to manage. Instead, the signal from the satellite connects to many remote locations at once, which is much more cost effective and efficient!

Metro Ethernet Metropolitan-area Ethernet is a metropolitan area network (MAN) that's based on Ethernet standards and can connect a customer to a larger network and the Internet. If available, businesses can use Metro Ethernet to connect their own offices together, which is another very cost-effective connection option. MPLS-based Metro Ethernet networks use MPLS in the ISP by providing an Ethernet or fiber cable to the customer as a connection. From the customer, it leaves the Ethernet cable, jumps onto MPLS, and then Ethernet again on the remote side. This is a smart and thrifty solution that's very popular if you can get it in your area.

Cable and DSL

Okay, before we get into talking about the principal serial encapsulation connections used on Cisco routers, like HDLC, PPP, and Frame Relay, I'm going to discuss cable modems and DSL, including ADSL and PPPoE, as solutions for connections to wide area networks. I think this approach will really help you understand the practical differences between DSL and cable modem networking.

DSL and cable Internet services actually do have a lot in common, but they still have some basic, essential differences that are important for you to understand:

Speed Most would say that cable is faster than DSL Internet, but cable doesn't always win the race in real-world use.

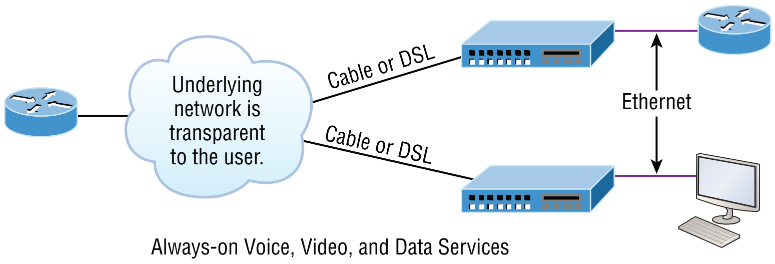
Security DSL and cable are based on different network security models, and until recently, cable has been the reputed loser in this contest. But now, it's pretty much a toss-up, and both offer adequate security that meets the needs of most users. But when I say adequate, I mean that there are still some very real security issues relating to both alternatives, no matter what your ISP says!

Popularity Cable Internet is definitely "best in show" in the United States, but DSL is beginning to catch up.

Customer satisfaction Here, the reverse is true—in the United States, DSL is top dog. But still, do you know anyone who's really totally satisfied with their ISP?

Figure 7.3 shows how a connection can terminate from modems to either a PC directly or to a router. Typically, your router would run DHCP on that interface as well as PPPoE. Both DSL and cable high-speed Internet services are available to millions of residential and business consumers worldwide, but in some areas, only one and sometimes even none of these services are available.

FIGURE 7.3 Broadband access using cable or DSL


Underlying network is transparent to the user.
Cable or DSL
Cable or DSL
Ethernet
Always-on Voice, Video, and Data Services

Surprisingly, some of the differences between DSL and cable modem have nothing to do with the actual technologies—it comes down to the individual ISP. All other things being equal issues like cost, reliability, and quality of customer support for both installation and maintenance issues vary significantly from one provider to the next.

Cable

Cable is a great cost-effective connection for a small office or home office that's commonly referred to as SOHO—yes, there really is an acronym for everything! And even in larger organizations, cable or even DSL can be great to have as a backup link.

Here are a few cable network terms:

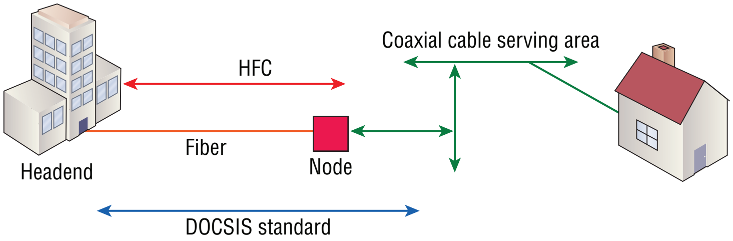
Headend This is where all cable signals are received, processed, and formatted. The signals are then transmitted over the distribution network from the headend.

Distribution network These are relatively small service areas that usually range in size from 100 to 2,000 customers. They're typically made up of a mixed, fiber-coaxial, or HFC architecture, with optical fiber substituting for the distribution network's trunk portion. The fiber forms both the connection from the headend and an optical node that changes light to radio frequency (RF) signals that are then distributed through a coaxial cable throughout the specific area being serviced.

Data Over Cable Service Interface Specification (DOCSIS) All cable modems and like devices have to measure up to this standard.

Figure 7.4 shows where you would find the various types of networks and how most of the terms I just listed would be used in a network diagram.

FIGURE 7.4 Cable network and terms



Let me explain the tangle here: ISPs often use a fiber-optic network that extends from the cable operator's master headend, sometimes even to regional headends, and proceeds out to a neighborhood's hub site before finally arriving at a fiber-optic node, which serves anywhere from 25 to 2,000 or more homes. Not exactly an elegant process, but don't get me wrong, all types of links have their own, specific problems—I'm not just picking on cable!

As if that wasn't enough, here's another issue: if you have cable, open your PC's command prompt and type **`ipconfig`** and then check out your subnet mask. It's probably a /20 or /21 Class B address. Oh my! You already know that means there are either 4,094 or 2,046 hosts per cable network connection, right? This level of population density makes Manhattan look rural and empty!

When we say "cable," we're really referring to the act of using coax (coaxial) cable for transmission. And CATV, or community antenna television, is now used as a means to offer cost-effective broadcasting to subscribers. So cable can provide voice and data, plus analog and digital video, all without devouring your whole paycheck.

Your average cable connection gives you a maximum download speed of 20 Mbps or more. And remember—you have to share that bandwidth with all those other subscribers. As if that weren't enough, there are other things like overloaded web servers and plain old Net congestion that factor in as well. But your email-checking neighbors really aren't making that much of a difference. So who or what is? Well, if you're an online gamer, you would likely notice a bit more lag during peak periods, which can be a matter of virtual life and death! And if somebody in your neighborhood is uploading a large amount of data like, say, an entire collection of *Star Wars* movies, well, that could definitely max out the entire connection, bringing everyone's browser to a crawl.

Cable modem access may or may not be faster or easier to install than DSL, and your mileage will vary, depending on where you live plus a variety of other factors. But it's usually more available and a tad less pricey, making it a winner by a nose. But no worries, if cable access isn't available in your neighborhood, DSL is okay—anything is better than dial-up!

Digital Subscriber Line (DSL)

Coming in second in our subscriber-based popularity contest is *digital subscriber line (DSL)*, a technology that uses your garden-variety copper phone wires to give you high-speed data transmission. DSL requires a phone line, a DSL modem that's usually included with the service, either an Ethernet card or a router that has an Ethernet connection, and someone that can provide service wherever you happen to be located.

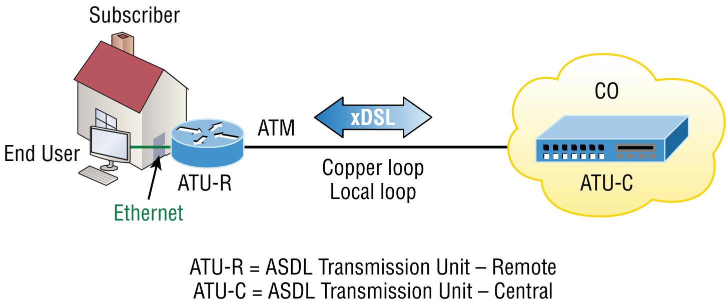
The acronym *DSL* originally meant "digital subscriber loop," but now its meaning has morphed to "line." DSL group types fall into two categories based upon upstream or downstream speed connections:

Symmetrical DSL The speed for both downstream and upstream connections are equal, or symmetrical. SDSL has a service distance of 12,000 feet at high speeds.

Asymmetrical DSL Different transmission speeds occur between two ends of a network—downstream speed is always faster. ADSL has a service distance of 18,000 feet at high speeds.

Figure 7.5 shows an average home user with xDSL, which is a transmission technology that moves data over copper pairs.

FIGURE 7.5 xDSL connection from home user to central office. All types of DSL are layer 1 technologies.


Subscriber
End User
Ethernet
ATU-R
ATM
xDSL
Copper loop
Local loop
CO
ATU-C
ATU-R = ASDL Transmission Unit – Remote
ATU-C = ASDL Transmission Unit – Central

The term *xDSL* covers a number of DSL variations, such as Asymmetrical DSL (ADSL), high-bit-rate DSL (HDSL), Rate Adaptive DSL (RADSL), Synchronous DSL (SDSL), ISDN DSL (IDSL), and very-high-data-rate DSL (VDSL).

DSL flavors that don't use the voice frequencies band, like ADSL and VDSL, allow DSL lines to carry both data and voice signals simultaneously. Others, like SDSL and IDSL, which occupy the complete frequency range, can carry only data. And by the way, the data service that the DSL connection gives you is always on.

The speed that DSL service can offer depends on how far you are from the central office (CO)—the closer the better. In fact, you can blaze at rates up to around 6.1 Mbps if you're physically close enough!

ADSL

Asymmetrical DSL (ADSL) supports both voice and data at the same time, but it was created to allot more bandwidth downstream than upstream because it's best for residential subscribers who usually need more downstream bandwidth for activities like downloading video, movies, music, online gaming, general surfing, and getting emails—some of which include sizeable attachments. ADSL will give you a downstream rate from 256 Kbps to 8 Mbps, but anything going upstream is only going to reach around 1.5 Mbps max.

Plain old telephone service (POTS) provides a channel for analog voice transmission and can transmit without a problem with ADSL over the same twisted-pair telephone line. Actually, depending on the type of ADSL, not just two but three information channels commonly utilize the same wiring simultaneously. This is why people can use a phone line and an ADSL connection at the same time and not affect either service.

ATM is the Data Link layer protocol typically used over the DSL layer 1 connection from the CPE and is terminated at what's known as the DSLAM—an ATM switch that contains DSL interface cards, or ATU-Cs. After ADSL connections meet their end at the DSLAM, it switches the data over an ATM network to something called an aggregation router—a layer 3 device where the subscriber's IP connection then expires.

You know by now how important encapsulation is, so as you've probably guessed, any IP packets over an ATM and DSL connection must provide it. This happens in one of three ways, depending on your interface type and the service provider's switch:

PPPoE I'll discuss this in detail in the next section.

RFC1483 routing RFC1483 describes two different methods for carrying connectionless network traffic over an ATM network: routed protocols and bridged protocols.

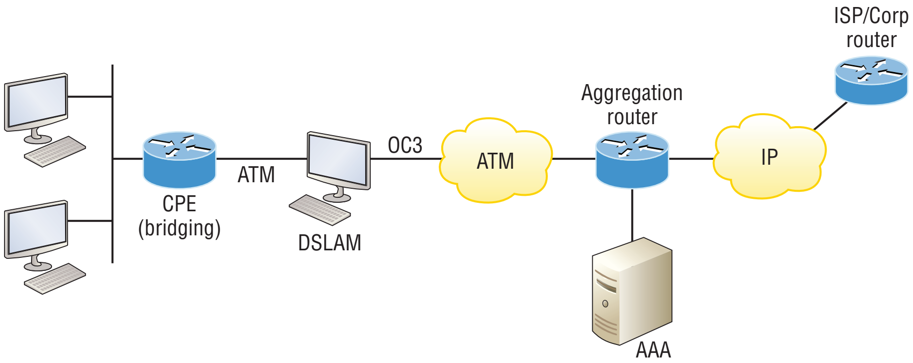
PPPoA Point-to-Point Protocol (PPP) over ATM is used to encapsulate PPP frames in ATM AAL5 (ATM Adaptation Layer 5). It's typically used with cable modems, DSL, and ADSL services and offers the usual PPP features of authentication, encryption, and compression. It actually has less overhead compared to PPPoE.

PPPoE

Used with ADSL services, PPPoE (Point-to-Point Protocol over Ethernet) encapsulates PPP frames in Ethernet frames and uses common PPP features like authentication, encryption, and compression. But as I said earlier, it's trouble if you've got a badly configured firewall. This is a tunneling protocol that layers IP and other protocols that run over PPP with the attributes of a PPP link so they can then be used to contact other Ethernet devices and initiate a point-to-point connection to transport IP packets.

Figure 7.6 displays typical usage of PPPoE over ADSL. As you can see, a PPP session is connected from the PC of the end user to the router and the subscriber PC IP address is assigned by the router via IPCP.

FIGURE 7.6 PPPoE with ADSL



PPPoE is used to equip custom PPP-based software with the ability to deal with a connection that's not using a serial line and to be at home in a packet-oriented network environment like Ethernet. It also allows for a custom connection with login and password for Internet connection accounting. Another factor is that the opposite side of the link's IP address is given to it and it's available only for the specific period that the PPPoE connection is open. This means that reusing IP addresses is dynamically permitted.

PPPoE has a discovery stage and a PPP session stage (see RFC 2516) that works like this: First, a host begins a PPPoE session, during which it has to execute a discovery process so it can choose the best server to meet the needs of the client machine's request. After that, it has to discover the Ethernet MAC address of the peer device and create a PPPoE session ID. So even though PPP delimits a peer-to-peer relationship, the discovery part is innately a client-server relationship.

Okay, before getting into serial connections, there's one last thing I want to cover—Cisco LRE.

Cisco Long Range Ethernet (LRE)

The Cisco Long Range Ethernet solution employs VDSL (very-high-data-rate digital subscriber line) technology to significantly expand Ethernet service capacity. And LRE can achieve impressive speeds from 5 to 15 Mbps (full-duplex) at distances up to 5,000 feet, traveling over existing twisted-pair wiring!

So basically, Cisco LRE technology can give us broadband service on POTS, digital telephone, and ISDN traffic lines, and it can also operate in modes compatible with ADSL technologies. This flexibility is important because it makes it possible for service providers to make LRE available in structures and/or buildings that have broadband services already in place but need it enhanced—very cool indeed!

Cabling the Serial Wide Area Network

As you can imagine, there are a few things that you need to know before connecting your WAN to ensure that everything goes well. For starters, you have to understand the kind of WAN Physical layer implementation that Cisco provides and be familiar with the various types of WAN serial connectors involved.

The good news is that Cisco serial connections support almost any type of WAN service. Your typical WAN connection is a dedicated leased line using HDLC, PPP, and Frame Relay with speeds that can kick it up to 45 Mbps (T3).

HDLC, PPP, and Frame Relay can use the same Physical layer specifications. I'll go over the various types of connections and then move on to tell you all about the WAN protocols specified in the CCNA R/S objectives.

Serial Transmission

WAN serial connectors use *serial transmission*, something that takes place 1 bit at a time over a single channel.

Older Cisco routers have used a proprietary 60-pin serial connector that you have to get from Cisco or a provider of Cisco equipment. Cisco also has a new, smaller proprietary serial connection that's about one-tenth the size of the 60-pin basic serial cable called the *smart-serial*. You have to verify that you have the right type of interface in your router before using this cable connector.

The type of connector you have on the other end of the cable depends on your service provider and its particular end-device requirements. There are several different types of ends you'll run into:

- EIA/TIA-232—Allowed speed up to 64 Kbps on 24-pin connector
- EIA/TIA-449
- V.35—Standard used to connect to a CSU/DSU, with speeds up to 2.048 Mbps using a 34-pin rectangular connector
- EIA-530

Make sure you're clear on these things: serial links are described in frequency, or cycles per second (hertz). The amount of data that can be carried within these frequencies is called *bandwidth*. Bandwidth is the amount of data in bits per second that the serial channel can carry.

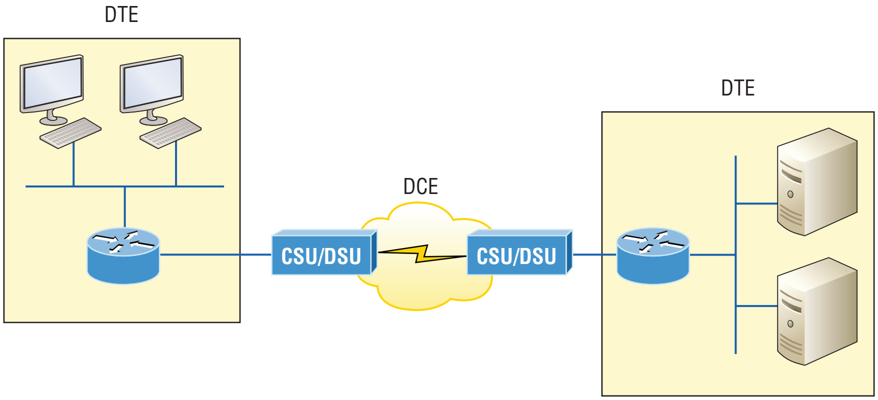
Data Terminal Equipment and Data Communication Equipment

By default, router interfaces are typically *data terminal equipment (DTE)*, and they connect into *data communication equipment (DCE)* like a *channel service unit/data service unit (CSU/DSU)* using a V.35 connector. CSU/DSU then plugs into a demarcation location (demarc) and is the service provider's last responsibility. Most of the time, the demarc is a jack that has an RJ45 (8-pin modular) female connector located in a telecommunications closet.

Actually, you may already have heard of demarcs. If you've ever had the glorious experience of reporting a problem to your service provider, they'll usually tell you everything tests out fine up to the demarc, so the problem must be the CPE, or customer premises equipment. In other words, it's your problem, not theirs!

Figure 7.7 shows a typical DTE-DCE-DTE connection and the devices used in the network.

FIGURE 7.7 DTE-DCE-DTE WAN connection: Clocking is typically provided by the DCE network to routers. In nonproduction environments, a DCE network is not always present.



The idea behind a WAN is to be able to connect two DTE networks through a DCE network. The DCE network includes the CSU/DSU, through the provider's wiring and switches, all the way to the CSU/DSU at the other end. The network's DCE device (CSU/DSU) provides clocking to the DTE-connected interface (the router's serial interface).

As mentioned, the DCE network provides clocking to the router; this is the CSU/DSU. If you have a nonproduction network and you're using a WAN crossover type of cable and do not have a CSU/DSU, then you need to provide clocking on the DCE end of the cable by using the `clock rate` command. To find out which interface needs the `clock rate` command, use the `show controllers` *`int`* command:

```
Corp#sh controllers s0/0/0
Interface Serial0/0/0
```

```
Hardware is PowerQUICC MPC860
DCE V.35, clock rate 2000000
```

The preceding output shows a DCE interface that has the clock rate set to `2000000`, which is the default for ISR routers. This next output shows a DTE connector, so you don't need enter the `clock rate` command on this interface:

```
SF#sh controllers s0/0/0
Interface Serial0/0/0
Hardware is PowerQUICC MPC860
DTE V.35 TX and RX clocks detected
```

NOTE

Terms such as *EIA/TIA-232*, *V.35*, *X.21*, and *HSSI (High-Speed Serial Interface)* describe the Physical layer between the DTE (router) and DCE device (CSU/DSU).

High-Level Data-Link Control (HDLC) Protocol

The High-Level Data-Link Control (HDLC) protocol is a popular ISO-standard, bit-oriented, Data Link layer protocol. It specifies an encapsulation method for data on synchronous serial data links using frame characters and checksums. HDLC is a point-to-point protocol used on leased lines. No authentication is provided by HDLC.

In byte-oriented protocols, control information is encoded using entire bytes. On the other hand, bit-oriented protocols use single bits to represent the control information. Some common bit-oriented protocols are SDLC and HDLC; TCP and IP are byte-oriented protocols.

HDLC is the default encapsulation used by Cisco routers over synchronous serial links. And Cisco's HDLC is proprietary, meaning it won't communicate with any other vendor's HDLC implementation. But don't give Cisco grief for it—*everyone's* HDLC implementation is proprietary. Figure 7.8 shows the Cisco HDLC format.

The reason every vendor has a proprietary HDLC encapsulation method is that each vendor has a different way for the HDLC protocol to encapsulate multiple Network layer protocols. If the vendors didn't have a way for HDLC to communicate the different layer 3 protocols, then HDLC would be able to operate in only a single layer 3 protocol environment. This proprietary header is placed in the data field of the HDLC encapsulation.

It's pretty simple to configure a serial interface if you're just going to connect two Cisco routers across a T1, for example. Figure 7.9 shows a point-to-point connection between two cities.

FIGURE 7.8 Cisco's HDLC frame format: Each vendor's HDLC has a proprietary data field to support multiprotocol environments.

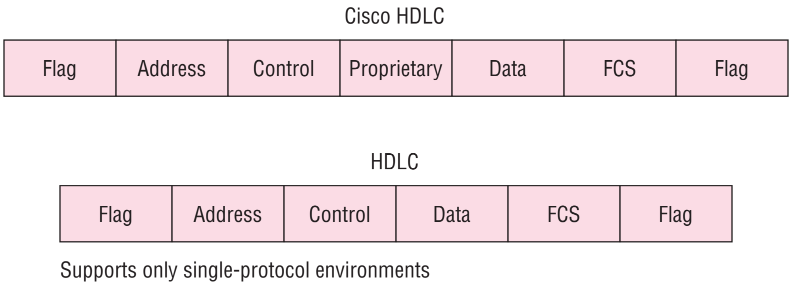
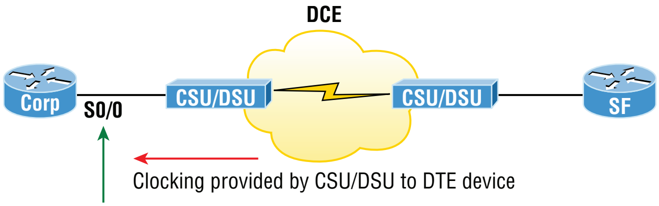

Cisco HDLC
Flag
Address
Control
Proprietary
Data
FCS
Flag
HDLC
Flag
Address
Control
Data
FCS
Flag
Supports only single-protocol environments

FIGURE 7.9 Configuring Cisco's HDLC proprietary WAN encapsulation


DCE
Corp
S0/0
CSU/DSU
CSU/DSU
SF
Clocking provided by CSU/DSU to DTE device

We can easily configure the routers with a basic IP address and then enable the interface. Assuming the link to the ISP is up, the routers will start communicating using the default HDLC encapsulation. Let's take a look at the Corp router configuration so you can see just how easy this can be:

```
Corp(config)#int s0/0
Corp(config-if)#ip address 172.16.10.1 255.255.255.252
Corp(config-if)#no shut


Corp#sh int s0/0
Serial0/0 is up, line protocol is up
  Hardware is PowerQUICC Serial
  Internet address is 172.16.10.1/30
  MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation HDLC, loopback not set
  Keepalive set (10 sec)
```

```
Corp#sh run | begin interface Serial0/0
interface Serial0/0
 ip address 172.16.10.1 255.255.255.252
!
```

Note that all I did was add an IP address before I then enabled the interface—pretty simple! Now, as long as the SF router is running the default serial encapsulation, this link will come up. Notice in the preceding output that the `show interface` command does show the encapsulation type of HDLC, but the output of `show running-config` does not. This is important—remember that if you don't see an encapsulation type listed under a serial interface in the active configuration file, you know it's running the default encapsulation of HDLC.

So let's say you have only one Cisco router and you need to connect to a non-Cisco router because your other Cisco router is on order or something. What would you do? You couldn't use the default HDLC serial encapsulation because it wouldn't work. Instead, you would need to go with an option like PPP, an ISO-standard way of identifying the upper-layer protocols. Now is a great time to get into more detail about PPP as well as how to connect to routers using the PPP encapsulation. You can check out RFC 1661 for more information on the origins and standards of PPP.

Point-to-Point Protocol (PPP)

Point-to-Point Protocol (PPP) is a Data Link layer protocol that can be used over either asynchronous serial (dial-up) or synchronous serial (ISDN) media. It relies on Link Control Protocol (LCP) to build and maintain data-link connections. Network Control Protocol (NCP) enables multiple Network layer protocols (routed protocols) to be used on a point-to-point connection.

Because HDLC is the default serial encapsulation on Cisco serial links and it works great, why in the world would you choose to use PPP? Well, the basic purpose of PPP is to transport layer 3 packets across a Data Link layer point-to-point link, and it's nonproprietary. So unless you have all Cisco routers, you need PPP on your serial interfaces because the HDLC encapsulation is Cisco proprietary, remember? Plus, since PPP can encapsulate several layer 3 routed protocols and provide authentication, dynamic addressing, and callback, PPP could actually be the best encapsulation solution for you over HDLC anyway.

Figure 7.10 shows the PPP protocol stack compared to the OSI reference model.

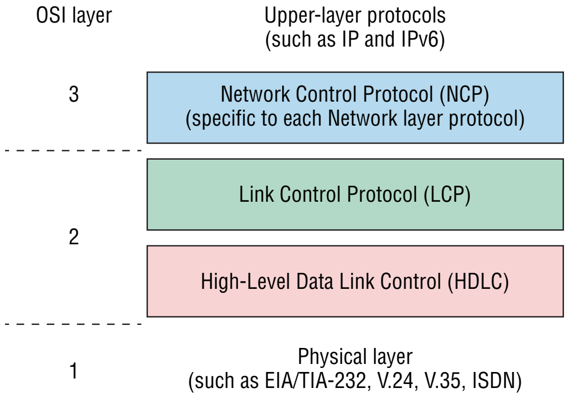
PPP contains four main components:

EIA/TIA-232-C, V.24, V.35, and ISDN A Physical layer international standard for serial communication.

HDLC A method for encapsulating datagrams over serial links.

LCP A method of establishing, configuring, maintaining, and terminating the point-to-point connection. It also provides features such as authentication. I'll give you a complete list of these features in the next section.

FIGURE 7.10 Point-to-Point Protocol stack


OSI layer
Upper-layer protocols
(such as IP and IPv6)
3
Network Control Protocol (NCP)
(specific to each Network layer protocol)
Link Control Protocol (LCP)
2
High-Level Data Link Control (HDLC)
1
Physical layer
(such as EIA/TIA-232, V.24, V.35, ISDN)

NCP A method of establishing and configuring different Network layer protocols for transport across the PPP link. NCP is designed to allow the simultaneous use of multiple Network layer protocols. Two examples of protocols here are Internet Protocol Control Protocol (IPCP) and Cisco Discovery Protocol Control Protocol (CDPCP).

Burn it into your mind that the PPP protocol stack is specified at the Physical and Data Link layers only. NCP is used to allow communication of multiple Network layer protocols by identifying and encapsulating the protocols across a PPP data link.

TIP

Remember that if you have a Cisco router and a non-Cisco router connected with a serial connection, you must configure PPP or another encapsulation method like Frame Relay because the HDLC default just won't work!

Next, we'll cover the options for LCP and PPP session establishment.

Link Control Protocol (LCP) Configuration Options

Link Control Protocol (LCP) offers different PPP encapsulation options, including the following:

Authentication This option tells the calling side of the link to send information that can identify the user. The two methods for this task are PAP and CHAP.

Compression This is used to increase the throughput of PPP connections by compressing the data or payload prior to transmission. PPP decompresses the data frame on the receiving end.

Error detection PPP uses Quality and Magic Number options to ensure a reliable, loop-free data link.

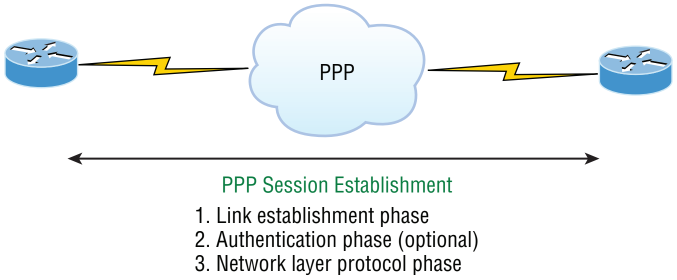
Multilink Starting with IOS version 11.1, multilink is supported on PPP links with Cisco routers. This option makes several separate physical paths appear to be one logical path at layer 3. For example, two T1s running multilink PPP would show up as a single 3 Mbps path to a layer 3 routing protocol.

PPP callback On a dial-up connection, PPP can be configured to call back after successful authentication. *PPP callback* can be a very good thing because it allows us to keep track of usage based upon access charges for accounting records and a bunch of other reasons. With callback enabled, a calling router (client) will contact a remote router (server) and authenticate. Predictably, both routers have to be configured for the callback feature for this to work. Once authentication is completed, the remote router will terminate the connection and then reinitiate a connection to the calling router from the remote router.

PPP Session Establishment

When PPP connections are started, the links go through three phases of session establishment, as shown in Figure 7.11:

FIGURE 7.11 PPP session establishment


PPP
PPP Session Establishment
1. Link establishment phase
2. Authentication phase (optional)
3. Network layer protocol phase

Link-establishment phase LCP packets are sent by each PPP device to configure and test the link. These packets contain a field called Configuration Option that allows each device to see the size of the data, the compression, and authentication. If no Configuration Option field is present, then the default configurations will be used.

Authentication phase If required, either CHAP or PAP can be used to authenticate a link. Authentication takes place before Network layer protocol information is read, and it's also possible that link-quality determination will occur simultaneously.

Network layer protocol phase PPP uses the *Network Control Protocol (NCP)* to allow multiple Network layer protocols to be encapsulated and sent over a PPP data link. Each Network layer protocol (e.g., IP, IPv6, which are routed protocols) establishes a service with NCP.

PPP Authentication Methods

There are two methods of authentication that can be used with PPP links:

Password Authentication Protocol (PAP) The *Password Authentication Protocol (PAP)* is the less secure of the two methods. Passwords are sent in clear text and PAP is performed only upon the initial link establishment. When the PPP link is first established, the remote node sends the username and password back to the originating target router until authentication is acknowledged. Not exactly Fort Knox!

Challenge Handshake Authentication Protocol (CHAP) The *Challenge Handshake Authentication Protocol (CHAP)* is used at the initial startup of a link and at periodic checkups on the link to ensure that the router is still communicating with the same host.

After PPP finishes its initial link-establishment phase, the local router sends a challenge request to the remote device. The remote device sends a value calculated using a one-way hash function called MD5. The local router checks this hash value to make sure it matches. If the values don't match, the link is immediately terminated.

NOTE CHAP authenticates at the beginning of the session and periodically throughout the session.

Configuring PPP on Cisco Routers

Configuring PPP encapsulation on an interface is really pretty straightforward. To configure it from the CLI, use these simple router commands:

```
Router#config t
Router(config)#int s0
Router(config-if)#encapsulation ppp
Router(config-if)#^Z
Router#
```

Of course, PPP encapsulation has to be enabled on both interfaces connected to a serial line in order to work and there are several additional configuration options available to you via the `ppp ?` command.

Configuring PPP Authentication

After you configure your serial interface to support PPP encapsulation, you can then configure authentication using PPP between routers. But first, you must set the hostname of the router if it hasn't been set already. After that, you set the username and password for the remote router that will be connecting to your router, like this:

```
Router#config t
Router(config)#hostname RouterA
RouterA(config)#username RouterB password cisco
```

When using the `username` command, remember that the username is the hostname of the remote router that's connecting to your router. And it's case sensitive too. Also, the password on both routers must be the same. It's a plain-text password that you can see with a `show run` command, and you can encrypt the password by using the command `service password-encryption`. You must have a username and password configured for each remote system you plan to connect to. The remote routers must also be similarly configured with usernames and passwords.

Now, after you've set the hostname, usernames, and passwords, choose either CHAP or PAP as the authentication method:

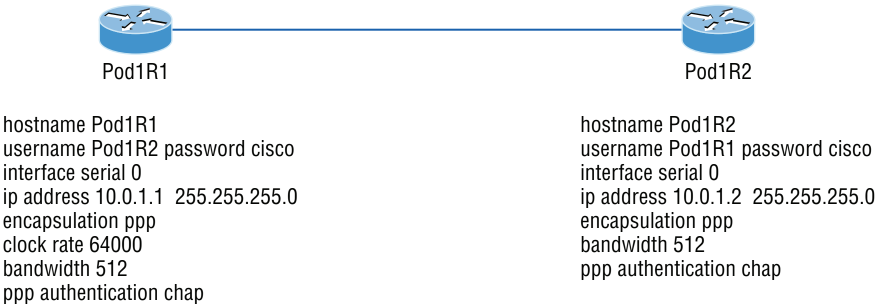
```
RouterA#config t
RouterA(config)#int s0
RouterA(config-if)#ppp authentication chap pap
RouterA(config-if)#^Z
RouterA#
```

If both methods are configured on the same line as I've demonstrated here, then only the first method will be used during link negotiation. The second acts as a backup just in case the first method fails.

Verifying and Troubleshooting Serial Links

Okay—now that PPP encapsulation is enabled, you need to verify that it's up and running. First, let's take a look at a figure of a sample nonproduction network serial link. Figure 7.12 shows two routers connected with a point-to-point serial connection, with the DCE side on the Pod1R1 router.

FIGURE 7.12 PPP authentication example



```
hostname Pod1R1
username Pod1R2 password cisco
interface serial 0
ip address 10.0.1.1  255.255.255.0
encapsulation ppp
clock rate 64000
bandwidth 512
ppp authentication chap
```

```
hostname Pod1R2
username Pod1R1 password cisco
interface serial 0
ip address 10.0.1.2  255.255.255.0
encapsulation ppp
bandwidth 512
ppp authentication chap
```

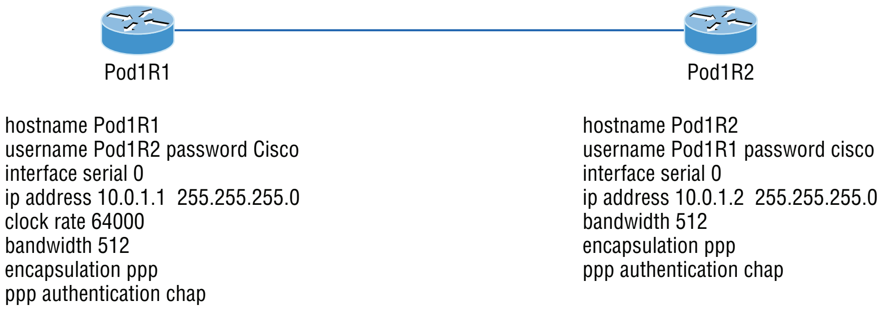
You can start verifying the configuration with the `show interface` command like this:

```
Pod1R1#sh int s0/0
Serial0/0 is up, line protocol is up
  Hardware is PowerQUICC Serial
  Internet address is 10.0.1.1/24
  MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec,
     reliability 239/255, txload 1/255, rxload 1/255
  Encapsulation PPP
  loopback not set
  Keepalive set (10 sec)
  LCP Open
  Open: IPCP, CDPCP
[output cut]
```

The first line of output is important because it tells us that serial 0/0 is up/up. Notice that the interface encapsulation is PPP and that LCP is open. This means that it has negotiated the session establishment and all is well. The last line tells us that NCP is listening for the protocols IP and CDP, shown with the NCP headers IPCP and CDPCP.

But what would you see if everything isn't so perfect? I'm going to type in the configuration shown in Figure 7.13 to find out.

FIGURE 7.13 Failed PPP authentication



```
hostname Pod1R1
username Pod1R2 password Cisco
interface serial 0
ip address 10.0.1.1  255.255.255.0
clock rate 64000
bandwidth 512
encapsulation ppp
ppp authentication chap
```

```
hostname Pod1R2
username Pod1R1 password cisco
interface serial 0
ip address 10.0.1.2  255.255.255.0
bandwidth 512
encapsulation ppp
ppp authentication chap
```

Okay—what's wrong here? Take a look at the usernames and passwords. Do you see the problem now? That's right, the *C* is capitalized on the Pod1R2 username command found in the configuration of router Pod1R1. This is wrong because the usernames and passwords are case sensitive. Now let's take a look at the `show interface` command and see what happens:

```
Pod1R1#sh int s0/0
Serial0/0 is up, line protocol is down
  Hardware is PowerQUICC Serial
  Internet address is 10.0.1.1/24
  MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec,
     reliability 243/255, txload 1/255, rxload 1/255
  Encapsulation PPP, loopback not set
  Keepalive set (10 sec)
  LCP Closed
  Closed: IPCP, CDPCP
```

First, notice that the first line of output shows us that `Serial0/0 is up` and `line protocol is down`. This is because there are no keepalives coming from the remote router. The next thing I want you to notice is that the LCP and NCP are closed because the authentication failed.

Debugging PPP Authentication

To display the CHAP authentication process as it occurs between two routers in the network, just use the command `debug ppp authentication`.

If your PPP encapsulation and authentication are set up correctly on both routers and your usernames and passwords are all good, then the `debug ppp authentication` command will display an output that looks like the following output, which is called the three-way handshake:

```
d16h: Se0/0 PPP: Using default call direction
1d16h: Se0/0 PPP: Treating connection as a dedicated line
1d16h: Se0/0 CHAP: O CHALLENGE id 219 len 27 from "Pod1R1"
1d16h: Se0/0 CHAP: I CHALLENGE id 208 len 27 from "Pod1R2"
1d16h: Se0/0 CHAP: O RESPONSE id 208 len 27 from "Pod1R1"
1d16h: Se0/0 CHAP: I RESPONSE id 219 len 27 from "Pod1R2"
1d16h: Se0/0 CHAP: O SUCCESS id 219 len 4
1d16h: Se0/0 CHAP: I SUCCESS id 208 len 4
```

But if you have the password wrong as they were previously in the PPP authentication failure example back in Figure 7.13, the output would look something like this:

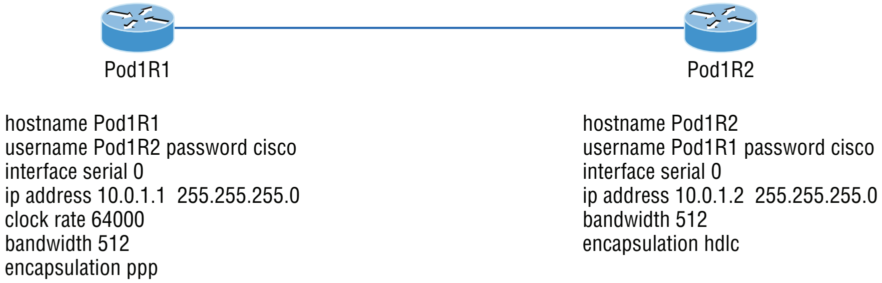
```
1d16h: Se0/0 PPP: Using default call direction
1d16h: Se0/0 PPP: Treating connection as a dedicated line
1d16h: %SYS-5-CONFIG_I: Configured from console by console
1d16h: Se0/0 CHAP: O CHALLENGE id 220 len 27 from "Pod1R1"
1d16h: Se0/0 CHAP: I CHALLENGE id 209 len 27 from "Pod1R2"
1d16h: Se0/0 CHAP: O RESPONSE id 209 len 27 from "Pod1R1"
1d16h: Se0/0 CHAP: I RESPONSE id 220 len 27 from "Pod1R2"
1d16h: Se0/0 CHAP: O FAILURE id 220 len 25 msg is "MD/DES compare failed"
```

PPP with CHAP authentication is a three-way authentication, and if the username and passwords aren't configured exactly the way they should be, then the authentication will fail and the link will go down.

Mismatched WAN Encapsulations

If you have a point-to-point link but the encapsulations aren't the same, the link will never come up. Figure 7.14 shows one link with PPP and one with HDLC.

FIGURE 7.14 Mismatched WAN encapsulations



```
hostname Pod1R1
username Pod1R2 password cisco
interface serial 0
ip address 10.0.1.1  255.255.255.0
clock rate 64000
bandwidth 512
encapsulation ppp
```

```
hostname Pod1R2
username Pod1R1 password cisco
interface serial 0
ip address 10.0.1.2  255.255.255.0
bandwidth 512
encapsulation hdlc
```

Look at router Pod1R1 in this output:

```
Pod1R1#sh int s0/0
Serial0/0 is up, line protocol is down
  Hardware is PowerQUICC Serial
  Internet address is 10.0.1.1/24
  MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec,
     reliability 254/255, txload 1/255, rxload 1/255
  Encapsulation PPP, loopback not set
  Keepalive set (10 sec)
  LCP REQsent
Closed: IPCP, CDPCP
```

The serial interface is up/down and LCP is sending requests but will never receive any responses because router Pod1R2 is using the HDLC encapsulation. To fix this problem, you would have to go to router Pod1R2 and configure the PPP encapsulation on the serial interface. One more thing: Even though the usernames are configured incorrectly, it doesn't matter because the command `ppp authentication chap` isn't used under the serial interface configuration. This means that the username command isn't relevant in this example.

You can set a Cisco serial interface back to the default of HDLC with the `no encapsulation` command like this:

```
Router(config)#int s0/0
Router(config-if)#no encapsulation
*Feb 7 16:00:18.678:%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/0,
changed state to up
```

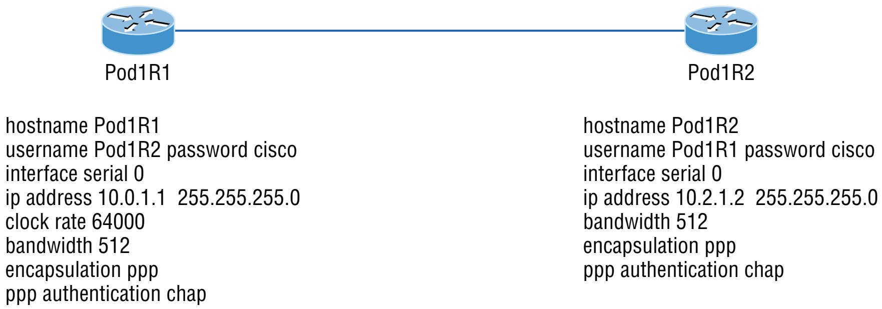
Notice the link came up because it now matches the encapsulation on the other end of the link!

NOTE Always remember that you just can't have PPP on one side and HDLC on the other—they don't get along!

Mismatched IP Addresses

A tricky problem to spot is if you have HDLC or PPP configured on your serial interface but your IP addresses are wrong. Things seem to be just fine because the interfaces will show that they are up. Take a look at Figure 7.15 and see if you can see what I mean—the two routers are connected with different subnets—router Pod1R1 with 10.0.1.1/24 and router Pod1R2 with 10.2.1.2/24.

FIGURE 7.15 Mismatched IP addresses



```
hostname Pod1R1
username Pod1R2 password cisco
interface serial 0
ip address 10.0.1.1  255.255.255.0
clock rate 64000
bandwidth 512
encapsulation ppp
ppp authentication chap
```

```
hostname Pod1R2
username Pod1R1 password cisco
interface serial 0
ip address 10.2.1.2  255.255.255.0
bandwidth 512
encapsulation ppp
ppp authentication chap
```

This will never work. Let's take a look at the output:

```
Pod1R1#sh int s0/0
Serial0/0 is up, line protocol is up
  Hardware is PowerQUICC Serial
  Internet address is 10.0.1.1/24
  MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation PPP, loopback not set
  Keepalive set (10 sec)
  LCP Open
  Open: IPCP, CDPCP
```

See that? The IP addresses between the routers are wrong but the link appears to be working just fine. This is because PPP, like HDLC and Frame Relay, is a layer 2 WAN encapsulation, so it doesn't care about IP addresses at all. So yes, the link is up, but you can't use IP across this link since it's misconfigured, or can you? Well, yes and no. If you try to ping you'll see that this actually works! This is a feature of PPP, but not HDLC or Frame Relay. But just because you can ping to an IP address that's not in the same subnet doesn't mean your network traffic and routing protocols will work. So be careful with this issue, especially when troubleshooting PPP links!

Take a look at the routing table of Pod1R1 and see if you can find the mismatched IP address problem:

```
[output cut]
  10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C       10.2.1.2/32 is directly connected, Serial0/0
C       10.0.1.0/24 is directly connected, Serial0/0
```

Interesting! We can see our serial interface S0/0 address of 10.0.1.0/24, but what is that other address on interface S0/0— 10.2.1.2/32? That's our remote router's interface IP address! PPP determines and places the neighbor's IP address in the routing table as a connected interface, which then allows you to ping it even though it's actually configured on a separate IP subnet.

TIP

For the Cisco objectives, you need to be able to troubleshoot PPP from the routing table as I just described.

To find and fix this problem, you can also use the `show running-config`, `show interfaces`, or `show ip interfaces brief` commands on each router, or you can use the `show cdp neighbors detail` command:

```
Pod1R1#sh cdp neighbors detail
-------------------------
Device ID: Pod1R2
Entry address(es):
  IP address: 10.2.1.2
```

Since the layer 1 Physical and layer 2 Data Link is up/up, you can view and verify the directly connected neighbor's IP address and then solve your problem.

Frame Relay

Frame Relay was one of the most popular WAN services deployed over the past two decades, and there's a good reason for this—cost. And it's a rare network design or designer that has the privilege to ignore that all-important cost factor!

By default, Frame Relay is classified as a nonbroadcast multi-access (NBMA) network, meaning it doesn't send any broadcasts such as RIP updates across the network.

In addition, Frame Relay is considerably more complex than the simple leased-line networks you learned about when we covered the HDLC and PPP protocols. The leased-line networks are easy to conceptualize, but not so much when it comes to Frame Relay. But with complexity comes versatility, which is why it's often represented as a "cloud" in networking graphics. I'll get to that more in a minute—for right now, I'm going to introduce Frame Relay in concept and explain how it differs from simpler leased-line technologies.

I'm going to begin your introduction to this technology by giving you a solid reference of all the new terms you'll need to really grasp the basics of Frame Relay. After that, I'll guide you through some simple Frame Relay implementations.

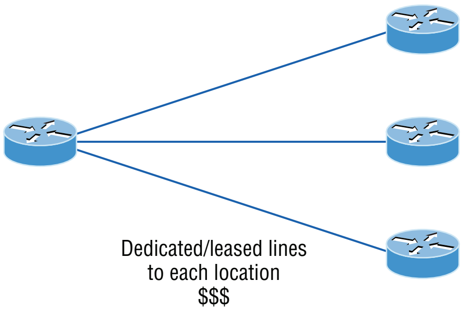
Introduction to Frame Relay Technology

As a CCNA R/S, you'll be expected to understand the basics of the Frame Relay technology and also to be able to configure it in simple scenarios. First, understand that Frame Relay is a packet-switched technology. From everything you've learned so far, just telling you this should make you immediately realize several things about it:

- You won't be using the `encapsulation hdlc` or `encapsulation ppp` command to configure it.
- Frame Relay doesn't work like a point-to-point leased line even though it can be made to look and act like one.
- Frame Relay is usually less expensive than leased lines are, but there are some sacrifices required to net that savings.

So why do we use Frame Relay? Figure 7.16 will help you form a snapshot of what a network looked like before Frame Relay as a first step to answering this question.

FIGURE 7.16 Before Frame Relay

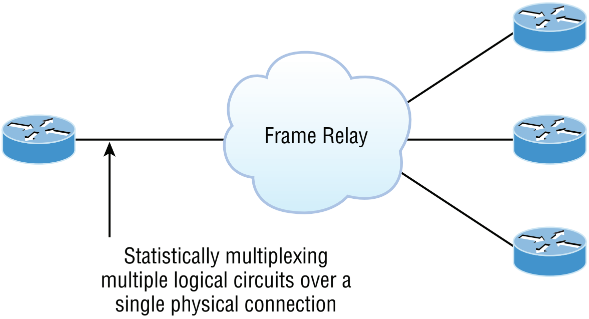


Now check out Figure 7.17. You can see that there's now only one connection between the Corporate router and the Frame Relay switch. That right there saves some major cash!

Let's say you needed to add seven remote sites to be accessed from the corporate office but you only have one free serial port on your router—it's Frame Relay to the rescue! Of course, this also means that you now have a single point of failure, which is not so good. But Frame Relay was typically used to save money, not to make a network more resilient.

Coming up, I'm going to cover the Frame Relay technology information you need to affectively meet the Cisco objectives.

FIGURE 7.17 After Frame Relay


Frame Relay
Statistically multiplexing multiple logical circuits over a single physical connection

Committed Information Rate (CIR)

Frame Relay provides a packet-switched network to lots of different customers at the same time, which is good because it spreads out the cost of the switches among many customers. But remember, Frame Relay is based on the assumption that those dependent on it will never need to transmit data constantly or simultaneously.

Frame Relay works by providing a portion of dedicated bandwidth to each user, and it also allows the user to exceed their guaranteed bandwidth if resources on the telco network happen to be available. So basically, Frame Relay providers allow customers to buy a lower amount of bandwidth than what they really use. There are two separate bandwidth specifications with Frame Relay:

Access rate The maximum speed at which the Frame Relay interface can transmit.

CIR The maximum bandwidth of data guaranteed to be delivered. It's the average amount that the service provider will allow you to transmit.

If these two values are equal, the Frame Relay connection will operate pretty much as a leased line would. But these values can also be set differently. To clarify this, suppose I've bought a T1 access rate, giving me 1.544 Mbps and a CIR of 256 Kbps. With this option, the first 256 Kbps of traffic I send is guaranteed to be delivered. Anything beyond that is called a "burst"—a transmission that exceeds my guaranteed 256 Kbps rate, which can be any amount up to my contracted T1 access rate. But what would happen if I were to send a combined committed burst that exceeds the MBR, or maximum burst rate? Most likely, my excess traffic will be dropped, but whether or not it's to be toast really depends on the subscription level of my particular service provider.

Of course, in a perfect world, this always works beautifully. But remember that little word *guaranteed*—as in guaranteed rate of 256 Kbps, to be exact? This means that any burst of data you send that exceeds your guaranteed 256 Kbps rate will be delivered on something called a "best effort" basis of delivery. Or maybe it won't—if your telco's equipment doesn't have the capacity to deliver it at the time you transmitted, then your frames will be discarded and the DTE will be notified. Timing is everything—you can scream data out at six times your guaranteed rate of 256 Kbps (T1) *only if* your telco has the capacity available on its equipment at the exact time you try to send. This is "oversubscription" in action!

NOTE The CIR is the rate, in bits per second, at which the Frame Relay switch agrees to transfer data.

Frame Relay Encapsulation Types

When configuring Frame Relay on Cisco routers, you need to specify it as an encapsulation on serial interfaces. And as I said earlier, you can't use HDLC or PPP with Frame Relay. When you configure Frame Relay, you must choose one of two encapsulation types—Cisco and IETF (Internet Engineering Task Force), as shown in the following output:

```
RouterA(config)#int s0
RouterA(config-if)#encapsulation frame-relay ?
  ietf  Use RFC1490 encapsulation
  <cr>
```

Unless you manually type in **`ietf`**, the default encapsulation option is Cisco, and predictably, it's what you want to go with when connecting two Cisco devices. You'd opt for the IETF-type encapsulation if you needed to connect a Cisco device to a non-Cisco device with Frame Relay. Whichever you choose, make sure the Frame Relay encapsulation is the same on both ends.

Virtual Circuits

Frame Relay operates using *virtual circuits* as opposed to the physical circuits that leased lines use. These virtual circuits are what link together the thousands of devices connected to the provider's "cloud." Frame Relay provides a virtual circuit between your two DTE devices, which makes them appear to be connected via an actual circuit. In reality, they're dumping their frames into a large, shared infrastructure. You never see the complexity of what's actually happening inside the cloud because you only have a virtual circuit.

And on top of all that, there are two types of virtual circuits—permanent and switched. Permanent virtual circuits (PVCs) are by far the most common type in use today. What "permanent" means here is that the telco creates the mappings inside its gear, and as long as you pay the bill, they'll remain in place.

Switched virtual circuits (SVCs) are more like a phone call. The virtual circuit is established when data needs to be transmitted. The virtual circuit is dismantled when the data transfer is complete.

Data Link Connection Identifiers (DLCIs)

Frame Relay PVCs are identified to DTE end devices by *Data Link Connection Identifiers (DLCIs)*. A Frame Relay service provider typically assigns DLCI values, which are used on Frame Relay interfaces to distinguish between different virtual circuits. Because many virtual circuits can be terminated on one multipoint Frame Relay interface, many DLCIs are often affiliated with it.

To clarify this, let's say you have a central HQ with three branch offices. If you were to connect each branch office to HQ using a T1, you would need three serial interfaces on your router at HQ, one for each T1. Simple, right? But suppose you use Frame Relay PVCs instead. You could have a T1 at each branch connected to a service provider and only a *single* T1 at HQ. There would be three PVCs on the single T1 at HQ, one going to each branch. And even though there's only a single interface and a single CSU/DSU, the three PVCs function as three separate circuits. Remember what I said about saving money? How much for two additional T1 interfaces and a pair of CSU/DSUs? Answer: A lot! So, why not just go ahead and ask for a percentage of the savings in your bonus?

Okay, before we go on, I want to define Inverse ARP (IARP) and discuss how it's used with DLCIs in a Frame Relay network. Yes, it is somewhat similar to ARP in the fact that it maps a DLCI to an IP address—kind of like ARP does with MAC addresses to IP addresses. And even though you can't configure IARP, you can disable it. It runs on a Frame Relay router and maps the DLCI to an IP address for Frame Relay so it knows how to get to the IP address at the other end of the PVC. You can see IP-to-DLCI mappings with the `show frame-relay map` command.

But if you have a non-Cisco router living in your network and it doesn't support IARP, then you're stuck with having to statically provide IP-to-DLCI mappings with the `frame-relay map` command—something I'll demonstrate in a bit.

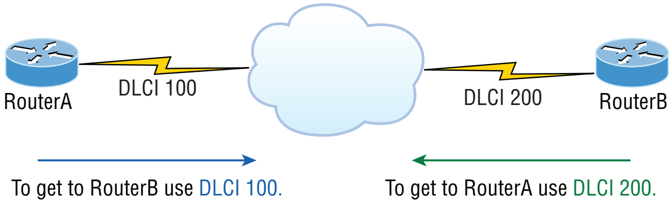
Inverse ARP (IARP) is used to map a known DLCI to an IP address.

Let's talk about DLCIs a bit more. They're locally significant—global significance requires the entire network to use the Local Management Interface (LMI) extensions that offer global significance. This is why you'll mostly find global DLCIs only in private networks.

But the DLCI doesn't have to be globally significant for it to be functional in getting a frame across the network. Let me explain: when RouterA wants to send a frame to RouterB, it looks up the IARP or manual mapping of the DLCI to the IP address it's trying to get to. Equipped with the DLCI, it then sends the frame out with the DLCI value it found in the DLCI field of the FR header. The provider's ingress switch gets this frame and does a lookup on the DLCI/physical-port combination it observes. Associated with that combination, it finds a new "locally significant" (meaning, between itself and the next-hop switch) DLCI to use in the header, and in the same entry in its table, it finds an outgoing physical port. This happens repeatedly all the way to RouterB. So basically, you could actually say that the DLCI for RouterA uses the entire virtual circuit to RouterB, even though each DLCI between every pair of devices could be completely different. The big point here is that RouterA is unaware of these differences. That's what makes the DLCI locally significant. So make a mental note that DLCIs really are used by the telco to "find" the other end of your PVC.

For a picture of why DLCIs are considered locally significant, take a look at Figure 7.18. In the figure, DLCI 100 is considered locally significant to RouterA and identifies the circuit to RouterB between RouterA and its ingress Frame Relay switch. DLCI 200 would identify this same circuit to RouterA between RouterB and its ingress Frame Relay switch.

FIGURE 7.18 DLCIs are local to your router. RouterA uses DLCI 100 to send data to RouterB.


RouterA
DLCI 100
DLCI 200
RouterB
To get to RouterB use DLCI 100.
To get to RouterA use DLCI 200.

DLCI numbers that are used to identify a PVC are typically assigned by the provider and start at 16.

You configure a DLCI number to be applied to a subinterface like this:

```
RouterA(config-if)#frame-relay interface-dlci ?
  <16-1007> Define a DLCI as part of the current subinterface
RouterA(config-if)#frame-relay interface-dlci 16
```

NOTE DLCIs identify the logical circuit between the local router and a Frame Relay switch.

Local Management Interface (LMI)

Local Management Interface (LMI) is a signaling standard used between your router and the first Frame Relay switch it's connected to. It allows for passing information about the operation and status of the virtual circuit between the provider's network and the DTE (your router). It communicates information about the following:

Keepalives These verify that data is flowing.

Multicasting This is an optional extension of the LMI specification that permits the efficient distribution of routing information and ARP requests over a Frame Relay network. Multicasting uses the reserved DLCIs from 1019 through 1022.

Global addressing This provides global significance to DLCIs, allowing the Frame Relay cloud to work exactly like a LAN. This has never been run in a production network to this day.

Status of virtual circuits This provides DLCI status. The status inquiries and messages are used as keepalives when there is no regular LMI traffic to send.

But remember, LMI is not communication between your routers; it's communication between your router and the nearest Frame Relay switch. So it's entirely possible that the router on one end of a PVC is actively receiving LMI while the router on the other end of the PVC is not. And of course, PVCs won't work with one end down, which clarifies the local nature of LMI communications.

There are three different types of LMI message formats: Cisco, ANSI, and Q.933A. The different kinds in use depend on both the type and configuration of the telco's switching gear, so it's imperative that you configure your router for the correct format, which should be provided by the telco.

NOTE Beginning with IOS version 11.2, the LMI type is autosensed. This enables the interface to determine the LMI type supported by the switch. If you're not going to use the autosense feature, you'll need to check with your Frame Relay provider to find out which type to use instead.

On Cisco equipment, the default type is, surprise, Cisco, but you still might have to change to ANSI or Q.933A, depending on what your service provider tells you. The three different LMI types are shown in the following router output:

```
RouterA(config-if)#frame-relay lmi-type ?
  cisco
  ansi
  q933a
```

As seen in the output, all three standard LMI signaling formats are supported. Here's a description of each one:

Cisco LMI defined by the Gang of Four (default). The Local Management Interface (LMI) was developed in 1990 by Cisco Systems, StrataCom, Northern Telecom, and Digital Equipment Corporation and became known as the Gang-of-Four LMI, or Cisco LMI.

ANSI Annex D included with ANSI standard T1.617.

ITU-T (Q.933A) Annex A included in the ITU-T standard, defined by using the `q933a` command keyword.

Routers receive LMI information from the service provider's Frame Relay switch on a frame-relay encapsulated interface and update the virtual circuit status to one of three different states:

Active state Everything is up, and routers can exchange information.

Inactive state The router's interface is up and working with a connection to the switching office, but the remote router isn't up.

Deleted state No LMI information is being received on the interface from the switch, which could be due to a mapping problem or a line failure.

Frame Relay Congestion Control

Remember back to our talk about CIR? From that, it should be obvious that the lower your CIR is set, the greater the risk that your data will be lost. This can be easily avoided if you have just one key piece of information—the most optimal time to transmit that huge burst! This begs the question: Is there any way for us to find out when our telco's shared infrastructure is free and clear and when it's crammed and jammed, just as we check the freeway before we leave work? And if so, how do we find out? Well, that's exactly what I'm going to talk about next—how the Frame Relay switch notifies the DTE of congestion problems!

Here are the three congestion bits and their meanings, as shown in the `show frame-relay map` output:

```
R1#sh frame map
[output cut]
input pkts 14055       output pkts 32795        in bytes 1096228
out bytes 6216155      dropped pkts 0           in FECN pkts 0
in BECN pkts 0         out FECN pkts 0          out BECN pkts 0
in DE pkts 0           out DE pkts 0
out bcast pkts 32795   out bcast bytes 6216155
```

Discard Eligibility (DE) As you know, when you transmit a burst of packets beyond the CIR of a given PVC, any packets exceeding the CIR could very well be discarded if the provider's network is congested at the time. Because of this, the excessive bits are marked with a *Discard Eligibility (DE)* bit in the Frame Relay header. And if the provider's network happens to be congested, the Frame Relay switch will discard the packets with the first DE bit set. So if your bandwidth is configured with a CIR of zero, the DE will always be on.

Forward Explicit Congestion Notification (FECN) When the Frame Relay network recognizes congestion in the cloud, the switch will set the *Forward Explicit Congestion Notification (FECN)* bit to 1 in a Frame Relay packet header. This will indicate to the destination DTE that the path the frame just traversed is congested.

Backward Explicit Congestion Notification (BECN) When the switch detects congestion in the Frame Relay network, it'll set the *Backward Explicit Congestion Notification (BECN)* bit in a Frame Relay frame that's destined for the source router. This notifies the originating router that congestion is ahead. But Cisco routers won't take action on this congestion information unless you tell them to!

NOTE

If your `in FECN` count is incrementing, the local PVC is congested. If your `in BECN` count is increasing, then the remote PVC is congested.

Troubleshooting Using Frame Relay Congestion Control

Now let's say all your users are whining about the fact that their Frame Relay connection to the corporate site is super slow. Because you strongly suspect that the link is overloaded, you verify the Frame Relay congestion control information with the `show frame-relay pvc` command and get this output:

```
RouterA#sh frame-relay pvc


PVC Statistics for interface Serial0/0 (Frame Relay DTE)
              Active     Inactive      Deleted       Static
  Local          1           0            0             0
  Switched       0           0            0             0
  Unused         0           0            0             0


DLCI = 100, DLCI USAGE = LOCAL, PVC STATUS = ACTIVE, INTERFACE = Serial0/0
  input pkts 1300          output pkts 1270       in bytes 21212000
  out bytes 21802000       dropped pkts 4         in pkts dropped 147
  out pkts dropped 0       out bytes dropped 0      in FECN pkts 147
   in BECN pkts 192        out FECN pkts 147
  out BECN pkts 259        in DE pkts 0             out DE pkts 214
  out bcast pkts 0         out bcast bytes 0
  pvc create time 00:00:06, last time pvc status changed 00:00:06
Pod1R1#
```

What you want to look for in this output is the `in BECN pkts 192` output because this is what's telling the local router that traffic sent to the corporate site is experiencing congestion. BECN means that the path that a frame took to get back to you is congested.

Frame Relay Implementation and Monitoring

Visit ccna .gg/ch7/a for a companion MicroNugget from CBT Nuggets.

As I've said, there are a ton of Frame Relay commands and configuration options, but I'm going to zero in on the ones you really need to know when studying for the CCNA R/S exam objectives. I'm going to start with one of the simplest configuration options—two routers with a single PVC between them. Next, I'll show you a more complex configuration using subinterfaces and demonstrate some of the monitoring commands available to verify the configuration.

Single Interface

Let's get started by looking at a simple example where we just want to connect two routers with a single PVC. Here's how that configuration would look:

```
RouterA#config t
RouterA(config)#int s0/0
RouterA(config-if)#encapsulation frame-relay
RouterA(config-if)#ip address 172.16.20.1 255.255.255.0
RouterA(config-if)#frame-relay lmi-type ansi
RouterA(config-if)#frame-relay interface-dlci 101
RouterA(config-if)#^Z
RouterA#
```

The first step is to specify the encapsulation as Frame Relay. Notice that since I didn't specify a particular encapsulation type—either Cisco or IETF—the Cisco default type was used. If the other router were non-Cisco, I would've specified IETF. Next, I assigned an IP address to the interface and then specified the LMI type of ANSI based on information provided by the telecommunications provider. Again, the default is Cisco. Finally, I added the DLCI of 101, which indicates the PVC we want to use and has been given to me by my ISP, assuming there's only one PVC on this physical interface.

That's all there is to it—if both sides are configured correctly, and the switch is working, the circuit will come up.

TIP

Check out Hands-on Lab 7.3 for a complete example of this type of configuration, including instructions on creating your own Frame Relay switch from a router.

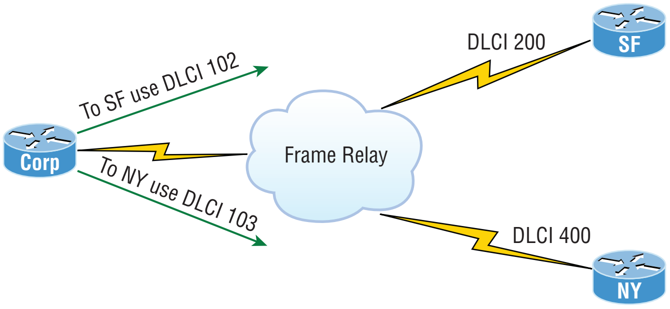
Subinterfaces

As I mentioned earlier, you can have multiple virtual circuits on a single serial interface and yet treat each as a separate interface. We can make this happen by creating *subinterfaces*. Think of a subinterface as a logical interface that's defined by the IOS software. Several subinterfaces will share a single hardware interface, yet for configuration purposes, they operate as if they were separate physical interfaces. This is known as multiplexing.

To configure a router in a Frame Relay network so it will avoid split horizon issues that will not permit certain routing updates, just configure a separate subinterface for each PVC with a unique DLCI and subnet assigned to the subinterface.

You define subinterfaces using a command like `int s0/0.`*`subinterface number`*. First, you have to set the encapsulation on the physical serial interface, and then you can define the subinterfaces—generally one subinterface per PVC. Here's an example, using Figure 7.19:

FIGURE 7.19 Configuring subinterfaces



```
Corp(config)#int s0/0
Corp(config-if)#no shut
Corp(config-if)#encapsulation frame-relay
Corp(config-if)#int s0/0.?
  <0-4294967295>  Serial interface number
Corp(config-if)#int s0/0.102 ?
  multipoint      Treat as a multipoint link
  point-to-point  Treat as a point-to-point link
Corp(config-if)#int s0/0.102 point-to-point
Corp(config-subif)#ip address 10.1.12.1 255.255.255.0
Corp(config-subif)#frame-relay interface-dlci 102
Corp(config-subif)#int s0/0.103
Corp(config-subif)#ip address 10.1.13.1 255.255.255.0
Corp(config-subif)#frame-relay interface-dlci 103
```

Make sure you don't have an IP address under the physical interface if you have configured subinterfaces!

You can define a legion of subinterfaces on any given physical interface, but keep in mind that there are only about a thousand available DLCIs. In the preceding example, I chose to use subinterface 102 and 103 because they represent the DLCI number assigned to that PVC by the carrier. There are two types of subinterfaces:

Point-to-point Used when a single virtual circuit connects one router to another. Each point-to-point subinterface requires its own subnet.

A point-to-point subinterface maps a single IP subnet per DLCI and addresses and resolves NBMA split horizon issues.

Multipoint This is when the router is the center of a star of virtual circuits that are using a single subnet for all routers' serial interfaces connected to the Frame Relay cloud. You'll usually find this implemented with the hub router in this mode with the spoke routers in physical interface (always point-to-point) or point-to-point subinterface mode.

In the following output, notice that the subinterface number matches the DLCI number—not a requirement, but it seriously helps you administer the interfaces:

```
interface Serial0/0
 no ip address (notice there is no IP address on the physical interface!)
 no ip directed-broadcast
 encapsulation frame-relay
!
interface Serial0/0.102 point-to-point
```

```
 ip address 10.1.12.1 255.255.255.0
 no ip directed-broadcast
frame-relay interface-dlci 102
!
interface Serial0/0.103 point-to-point
 ip address 10.1.13.1 255.255.255.0
 no ip directed-broadcast
frame-relay interface-dlci 103
!
```

Notice that there's no LMI type defined. This means that the routers are either running the Cisco default or using autodetect if you're running Cisco IOS version 11.2 or newer. I also want to point out that each interface maps to a single DLCI and is defined as a separate subnet. And remember—point-to-point subinterfaces solve split horizon issues as well!

Monitoring Frame Relay

Several commands are used frequently to check the status of your interfaces and PVCs once you have Frame Relay encapsulation set up and running. To list them, use the `show frame ?` command, as seen here:

```
Corp>sho frame ?
end-to-end     Frame-relay end-to-end VC information
fragment       show frame relay fragmentation information
ip             show frame relay IP statistics
lapf           show frame relay lapf status/statistics
lmi            show frame relay lmi statistics
map            Frame-Relay map table
pvc            show frame relay pvc statistics
qos-autosense  show frame relay qos-autosense information
route          show frame relay route
svc            show frame relay SVC stuff
traffic        Frame-Relay protocol statistics
vofr           Show frame-relay VoFR statistics
```

The most common parameters that you view with the `show frame-relay` command are `lmi`, `pvc`, and `map`.

Now, let's take a look at the most frequently used commands and the information they provide.

The *show frame-relay lmi* Command

The `show frame-relay lmi` command will give you the LMI traffic statistics exchanged between the local router and the Frame Relay switch. Here's an example:

```
Corp#sh frame lmi
```

```
LMI Statistics for interface Serial0/0 (Frame Relay DTE)LMI TYPE = CISCO
  Invalid Unnumbered info 0     Invalid Prot Disc 0
  Invalid dummy Call Ref 0      Invalid Msg Type 0
  Invalid Status Message 0      Invalid Lock Shift 0
  Invalid Information ID 0      Invalid Report IE Len 0
  Invalid Report Request 0      Invalid Keep IE Len 0
  Num Status Enq. Sent 61        Num Status msgs Rcvd 0
  Num Update Status Rcvd 0      Num Status Timeouts 60
```

The router output from the `show frame-relay lmi` command shows you any LMI errors, plus the LMI type. So, I have a question based on the output of the command. Is this Frame Relay network working? The answer is no because the router has sent 60 inquiries and has not received even one reply from the Frame-Relay switch. If you see this, you need to call the provider because this is a Frame Relay switch configuration issue.

The *show frame pvc* Command

The `show frame pvc` command will present you with a list of all configured PVCs and DLCI numbers. It provides the status of each PVC connection and traffic statistics too. It will also give you the number of BECN, FECN, and DE packets sent and received on the router per PVC.

Here is an example:

```
Corp#sho frame pvc


PVC Statistics for interface Serial0/0 (Frame Relay DTE)


DLCI = 102,DLCI USAGE = LOCAL,PVC STATUS =ACTIVE,
INTERFACE = Serial0/0.102
 input pkts 50977876    output pkts 41822892
  in bytes 3137403144
 out bytes 3408047602   dropped pkts 5
  in FECN pkts 0
 in BECN pkts 0      out FECN pkts 0     out BECN pkts 0
 in DE pkts 9393     out DE pkts 0
 pvc create time 7w3d, last time pvc status changed 7w3d


DLCI = 103,DLCI USAGE =LOCAL,PVC STATUS =ACTIVE,
INTERFACE = Serial0/0.103
 input pkts 30572401   output pkts 31139837
  in bytes 1797291100
 out bytes 3227181474   dropped pkts 5
  in FECN pkts 0
 in BECN pkts 0      out FECN pkts 0     out BECN pkts 0
 in DE pkts 28      out DE pkts 0
 pvc create time 7w3d, last time pvc status changed 7w3d
```

If you want to see information about only PVC 102, you can type the command **`show frame-relay pvc 102`**. Let's take a closer look at the output of this one line:

```
DLCI = 102,DLCI USAGE = LOCAL,PVC STATUS =ACTIVE,
INTERFACE = Serial0/0.102
```

The PVC status field in the output of the `show frame-relay pvc` command reports the status of the PVC between the router and the Frame-Relay switch. The switch (DCE) reports the status to the router (DTE) using the LMI protocol. There are three types of reported statuses:

ACTIVE The switch is correctly programmed with the DLCI and there is a successful DTE-to-DTE circuit (router to router).

INACTIVE The router is connected to the switch (DTE to DCE), but there's not a connection to the far-end router (DTE). This can be a router or switch configuration issue.

DELETED The router (DTE) is configured for a DLCI that the switch (DCE) does not recognize or is not configured correctly.

TIP The three LMI reported statuses are CCNA R/S objectives! Understand why you'd see each status.

The *show interface* Command

You can use the `show interface` command to check for LMI traffic. The `show interface` command displays information about the encapsulation as well as layer 2 and layer 3 information. It also displays line, protocol, DLCI, and LMI information. Check it out:

```
Corp#sho int s0/0
Serial0/0 is up, line protocol is up
 Hardware is HD64570
 MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec, rely
  255/255, load 2/255
 Encapsulation FRAME-RELAY, loopback not set, keepalive
  set (10 sec)
 LMI enq sent 451751,LMI stat recvd 451750,LMI upd recvd
  164,DTE LMI up
 LMI enq recvd 0, LMI stat sent 0, LMI upd sent 0
 LMI DLCI 1023 LMI type is CISCO frame relay DTE
 Broadcast queue 0/64, broadcasts sent/dropped 0/0,
  interface broadcasts 839294
```

The `LMI DLCI` is used to define the type of LMI being used. If it happens to be 1023, it's the default LMI type for Cisco routers. If the `LMI DLCI` is zero, then it's the ANSI LMI type (Q.933A uses 0 as well). If the `LMI DLCI` is anything other than 0 or 1023, it's a 911—call your provider; they've got major issues!

The *show frame map* Command

The `show frame map` command displays the mappings from the Network layer to DLCI. Here's how that looks, using Figure 7.19:

```
Corp#show frame map
Serial0/0.102 (up): ip 10.1.12.2 dlci 102(0x66,0x400),
              dynamic, broadcast,, status defined, active
Serial0/0.103 (up): ip 10.1.13.2 dlci 103(0x67,0x410),
              dynamic, broadcast,, status defined, active
```

Notice that the Network layer addresses are marked dynamic, which means they were resolved with the dynamic protocol Inverse ARP (IARP). After the DLCI number is listed, you can see some numbers in parentheses. The first one is 0x66, which is the hex equivalent for the DLCI number 102, used on serial 0/0.102. And the 0x67 is the hex for DLCI 103 used on serial 0/0.103. The second numbers, 0x400 and 0x410, are the DLCI numbers configured in the Frame Relay frame. They're different because of the way the bits are spread out in the frame.

Again, looking at Figure 7.19, the preceding output is telling the Corp router that to get SF using 10.1.12.2, use DLCI 102. To get to the NY router with IP address 10.1.13.2, use DLCI 103. The Corp router would never use a remote DLCI.

NOTE You must be able to find the DLCI number used to get to a remote site by using the `show frame-relay map` command.

The *debug frame lmi* Command

The `debug frame lmi` command will show real-time output on the router consoles by default (as with any `debug` command). The information this command gives you will enable you to verify and troubleshoot the Frame Relay connection by helping you determine whether the router and switch are exchanging the correct LMI information. Here's an example:

```
Corp#debug frame-relay lmi
Serial3/1(in): Status, myseq 214
RT IE 1, length 1, type 0
KA IE 3, length 2, yourseq 214, myseq 214
PVC IE 0x7 , length 0x6 , dlci 130, status 0x2 , bw 0
Serial3/1(out): StEnq, myseq 215, yourseen 214, DTE up
datagramstart = 0x1959DF4, datagramsize = 13
```

```
FR encap = 0xFCF10309
00 75 01 01 01 03 02 D7 D6


Serial3/1(in): Status, myseq 215
RT IE 1, length 1, type 1
KA IE 3, length 2, yourseq 215, myseq 215
Serial3/1(out): StEnq, myseq 216, yourseen 215, DTE up
datagramstart = 0x1959DF4, datagramsize = 13
FR encap = 0xFCF10309
00 75 01 01 01 03 02 D8 D7
```

Visit ccna .gg/ch7/b for a companion MicroNugget from CBT Nuggets.

Troubleshooting Frame Relay Networks

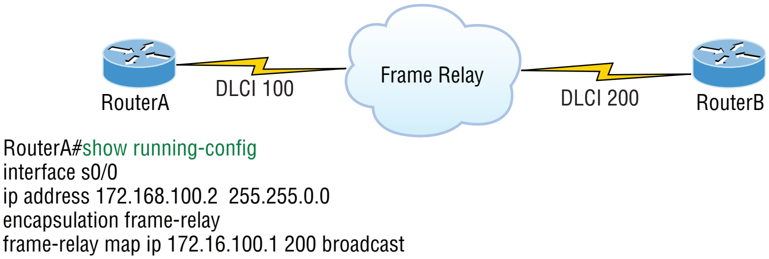
Troubleshooting Frame Relay networks isn't any harder than troubleshooting any other type of network as long as you know what to look for, which is what I'm going to cover now. I'll go over some basic problems that commonly occur in Frame Relay configuration and how to solve them.

First on the list are serial encapsulation problems. As you learned recently, there are two Frame Relay encapsulations: Cisco and IETF. Cisco is the default, and it means that you have a Cisco router on each end of the Frame Relay network. Using Figure 7.20, if you don't have a Cisco router on the remote end of your Frame Relay network, then you need to run the IETF encapsulation as shown here:

```
RouterA(config)#int s0
RouterA(config-if)#encapsulation frame-relay ?
  ietf  Use RFC1490 encapsulation
  <cr>
RouterA(config-if)#encapsulation frame-relay ietf
```

Once you verify that you're using the correct encapsulation, you then need to check out your Frame Relay mappings. For an example, take a look at Figure 7.20.

FIGURE 7.20 Frame Relay mappings

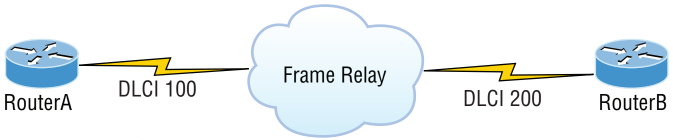


```
RouterA#show running-config
interface s0/0
ip address 172.168.100.2  255.255.0.0
encapsulation frame-relay
frame-relay map ip 172.16.100.1 200 broadcast
```

So why can't RouterA talk to RouterB across the Frame Relay network in Figure 7.20? To find that out, take a close look at the `frame-relay map` statement. See the problem now? You cannot use a remote DLCI to communicate to the Frame Relay switch; you must use *your* DLCI number! The mapping should have included DLCI 100 instead of DLCI 200.

Now that you know how to ensure that you have the correct Frame Relay encapsulation, and that DLCIs are only locally significant, let's look into some routing protocol problems that are often associated with Frame Relay. See if you can find a problem with the two configurations in Figure 7.21.

FIGURE 7.21 Frame Relay routing problems



```
RouterA#show running-config
interface s0/0
ip address 172.16.100.2  255.255.0.0
encapsulation frame-relay
frame-relay map ip 172.16.100.1 100
router rip
network 172.16.0.0
```

```
RouterB#show running-config
interface s0/0
ip address 172.16.100.1  255.255.0.0
encapsulation frame-relay
frame-relay map ip 172.16.100.2 200
router rip
network 172.16.0.0
```

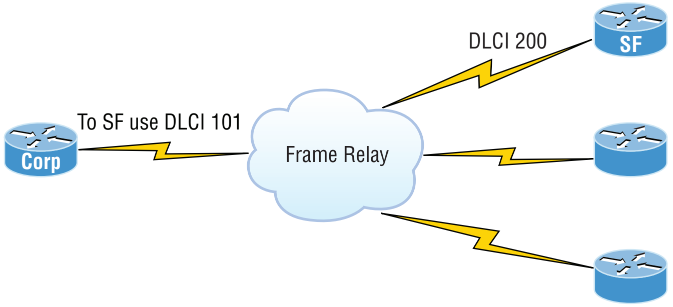
Hmmmm, the configs look pretty good. Actually, they look great, so what's the problem? Well, remember that Frame Relay is a nonbroadcast multi-access (NBMA) network by default, meaning that it doesn't allow any broadcasts across the PVC. So, because the mapping statements do not have the `broadcast` argument at the end of the line, broadcasts such as RIP updates or multicasts to neighbors such as Hello packets, won't be sent across the PVC. The correct line for RouterA would look like this:

```
frame-relay map ip 172.16.100.1 100 broadcast
```

But wait, do we even use RIP in our internetworks today? Maybe there are problems with the routing protocols EIGRP and OSPF over Frame Relay as well? Let's take a look at Figure 7.22 and see if you can spot a problem with the OSPF configuration. After this, we'll move on to troubleshooting EIGRP.

Since Frame Relay NBMA networks won't allow broadcasts or multicasts, an OSPF router will not attempt to dynamically discover any OSPF neighbors on the Frame Relay interface. Also, since this means that elections won't be allowed, you'd have to statically configure OSPF neighbors, plus the Corp router would need to be configured as a DR. Even though these are serial links, an NBMA network behaves like Ethernet and a DR is needed to exchange routing information. Only the Corp router can act as a DR because it would have the PVCs for all other routers. But the easiest way to fix this problem is to use the command `ip ospf network point-to-multipoint` on all router Frame Relay interfaces—not just the Corp router, but all branches too!

FIGURE 7.22 Frame Relay OSPF routing problems

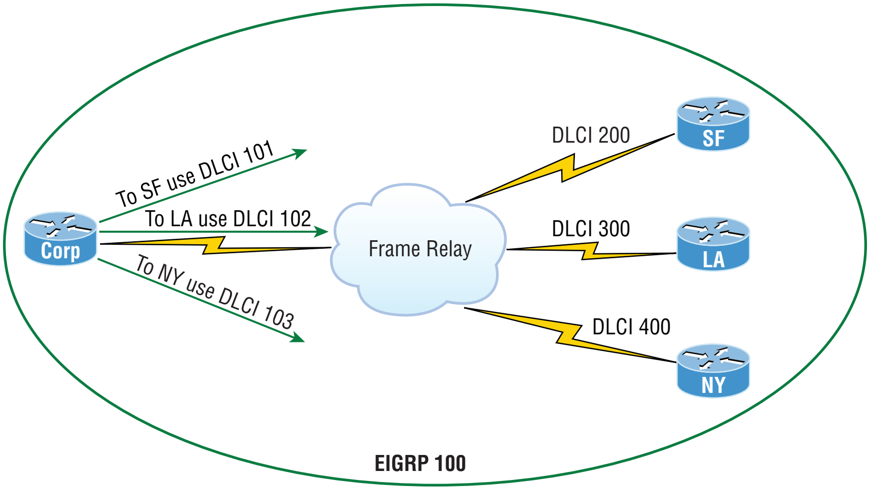

DLCI 200
SF
To SF use DLCI 101
Corp
Frame Relay

```
Corp#show running-config
interface s0/0
ip address 172.16.100.2  255.255.0.0
encapsulation frame-relay
frame-relay interface-dlci 101
frame-relay map ip 172.16.100.1 101
router ospf 1
network 172.16.0.0 0.0.255.255 area 0
```

```
SF#show running-config
interface s0/0
ip address 172.16.100.1  255.255.0.0
encapsulation frame-relay
frame-relay map ip 172.16.100.2 200
router ospf 1
network 172.16.0.0 0.0.255.255 area 0
```

Okay, this would solve the problem if you're running OSPF, but what if you're running EIGRP? In Figure 7.23 you can see three remote connection sites to the Corp router with all routers running EIGRP. The hosts behind the Corp router can communicate to all hosts in all remote networks, but hosts in SF, LA, and NY cannot communicate to each other.

FIGURE 7.23 Frame Relay EIGRP routing problems


To SF use DLCI 101
To LA use DLCI 102
To NY use DLCI 103
Corp
Frame Relay
DLCI 200
SF
DLCI 300
LA
DLCI 400
NY
EIGRP 100

Let's take a look at the configuration of the Corp router now:

```
interface Serial0/0
 ip address 192.168.10.1 255.255.255.0
 encapsulation frame-relay
frame-relay interface-dlci 101
frame-relay interface-dlci 102
frame-relay interface-dlci 103
!
```

The Frame Relay network is all on one subnet, and the configuration looks good, so why can't hosts on the remote networks communicate to each other? Here's your answer: The SF router sends an EIGRP route update to the Corp router and the Corp router updates the local routing table with a route to SF's network. LA and NY do the same thing and then each site's remote networks can communicate to the hosts behind the Corp router. However, when the Corp router sends route updates to the SF, LA, and NY routers, the updates never included each other's remote network because of the split horizon rule. This rules says you cannot advertise a network back out the same interface you received it on, which is the default configuration of all Cisco serial interfaces. This prevents the threat of network loops from occurring.

We can solve this problem with subinterfaces. Take a look at the Corp router's new configuration with subinterfaces, which solves the split horizon issues. Here are the steps to take:

1. Remove the IP address and `interface-dlci` commands from under the physical interface.
2. Create a subinterface (logical interface) for each PVC.
3. Design and implement a separate subnet (address space) for each subinterface.
4. Add the command `frame-relay interface-dlci` *`dlci`* under each subinterface.

```
interface Serial0/0
 no ip address (notice there is no IP address on the physical interface!)
 encapsulation frame-relay
!
interface Serial0/0.101 point-to-point
 ip address 192.168.10.1 255.255.255.252
frame-relay interface-dlci 101
!
interface Serial0/0.102 point-to-point
 ip address 192.168.10.5 255.255.255.252
frame-relay interface-dlci 102
!
interface Serial0/0.103 point-to-point
ip address 192.168.10.9 255.255.255.252
frame-relay interface-dlci 103
!
```

Notice that there is no IP address under the physical interface, that each subinterface is a separate subnet or address space, and that I needed to add the `frame-relay interface-dlci` command under each subinterface. Our split horizon issue is now resolved.

Virtual Private Networks

I'd be pretty willing to bet you've heard the term *VPN* more than once before. Maybe you even know what one is, but just in case, a *virtual private network (VPN)* allows the creation of private networks across the Internet, enabling privacy and tunneling of non-TCP/IP protocols. VPNs are used daily to give remote users and disjointed networks connectivity over a public medium like the Internet instead of using more expensive permanent means.

No worries—VPNs aren't really that hard to understand. A VPN fits somewhere between a LAN and WAN, with the WAN often simulating a LAN link because your computer, on one LAN, connects to a different, remote LAN and uses its resources remotely. The key drawback to using VPNs is a big one—security! So the definition of connecting a LAN (or VLAN) to a WAN may sound the same as using a VPN, but a VPN is actually much more.

Here's the difference: A typical WAN connects two or more remote LANs together using a router and someone else's network, like, say, your Internet service provider (ISP). Your local host and router see these networks as remote networks and not as local networks or local resources. This would be a WAN in its most general definition. A VPN actually makes your local host part of the remote network by using the WAN link that connects you to the remote LAN. The VPN will make your host appear as though it's actually local on the remote network. This means that we now have access to the remote LAN's resources, and that access is also very secure!

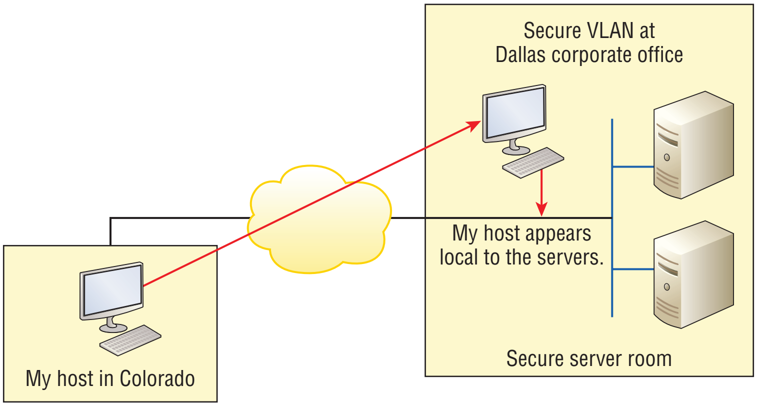
This may sound a lot like a VLAN definition, and really, the concept is the same: "Take my host and make it appear local to the remote resources." Just remember this key distinction: For networks that are physically local, using VLANs is a good solution, but for physically remote networks that span a WAN, opt for using VPNs instead.

For a simple VPN example, let's use my home office in Boulder, Colorado. Here, I have my personal host, but I want it to appear as if it's on a LAN in my corporate office in Dallas, Texas, so I can get to my remote servers. VPN is the solution I would opt for to achieve my goal.

Figure 7.24 shows this example of my host using a VPN connection from Boulder to Dallas, which allows me to access the remote network services and servers as if my host were right there on the same VLAN as my servers.

Why is this so important? If you answered, "Because my servers in Dallas are secure, and only the hosts on the same VLAN are allowed to connect to them and use the resources of these servers," you nailed it! A VPN allows me to connect to these resources by locally attaching to the VLAN through a VPN across the WAN. The other option is to open up my network and servers to everyone on the Internet or another WAN service, in which case my security goes "poof." So clearly, it's imperative I have a VPN!

FIGURE 7.24 Example of using a VPN



Benefits of VPNs

There are many benefits to using VPNs on your corporate and even home network. The benefits covered in the CCNA R/S objectives are as follows:

Security VPNs can provide very good security by using advanced encryption and authentication protocols, which will help protect your network from unauthorized access. IPsec and SSL fall into this category. Secure Sockets Layer (SSL) is an encryption technology used with web browsers, which has native SSL encryption, and is known as Web VPN. You can also use the Cisco AnyConnect SSL VPN client installed on your PC to provide an SSL VPN solution, as well as the Clientless Cisco SSL VPN.

Cost Savings By connecting the corporate remote offices to their closest Internet provider, and then creating a VPN tunnel with encryption and authentication, I gain a huge savings over opting for traditional leased point-to-point lines. This also permits higher bandwidth links and security, all for far less money than traditional connections.

Scalability VPNs scale very well to quickly bring up new offices or have mobile users connect securely while traveling or when connecting from home.

Compatibility with broadband technology For remote and traveling users and remote offices, any Internet access can provide a connection to the corporate VPN. This allows users to take advantage of the high-speed Internet access of DSL or cable modems.

VPNs are categorized based upon the role they play in a business. There are three different categories of VPNs:

> *Remote access VPNs* allow remote users such as telecommuters to securely access the corporate network wherever and whenever they need to.

Site-to-site VPNs, or intranet VPNs, allow a company to connect its remote sites to the corporate backbone securely over a public medium like the Internet instead of requiring more expensive WAN connections like Frame Relay.

Extranet VPNs allow an organization's suppliers, partners, and customers to be connected to the corporate network in a limited way for business-to-business (B2B) communications.

Now you're interested, huh? And since VPNs are inexpensive and secure, I'm guessing you just can't wait to find out how to create VPNs now! There's more than one way to bring a VPN into being. The first approach uses IPsec to create authentication and encryption services between endpoints on an IP network. The second way is via tunneling protocols, which allow you to establish a tunnel between endpoints on a network. And understand that the tunnel itself is a means for data or protocols to be encapsulated inside another protocol—pretty clean!

I'm going to go over IPsec in a minute, but first I really want to describe four of the most common tunneling protocols in use today:

Layer 2 Forwarding (L2F) is a Cisco-proprietary tunneling protocol, and it was Cisco's first tunneling protocol created for virtual private dial-up networks (VPDNs). A VPDN allows a device to use a dial-up connection to create a secure connection to a corporate network. L2F was later replaced by L2TP, which is backward compatible with L2F.

Point-to-Point Tunneling Protocol (PPTP) was created by Microsoft and others to allow the secure transfer of data from remote networks to the corporate network.

Layer 2 Tunneling Protocol (L2TP) was created by Cisco and Microsoft to replace L2F and PPTP. L2TP merged the capabilities of both L2F and PPTP into one tunneling protocol.

Generic Routing Encapsulation (GRE) is another Cisco-proprietary tunneling protocol. It forms virtual point-to-point links, allowing for a variety of protocols to be encapsulated in IP tunnels. I'll cover GRE in more detail, including how to configure it, at the end of this chapter.

Okay—now that you're clear on both exactly what a VPN is and the various types of VPNs available, it's time to dive into IPsec.

Introduction to Cisco IOS IPsec

Simply put, IPsec is an industry-wide standard framework of protocols and algorithms that allows for secure data transmission over an IP-based network and functions at the layer 3 Network layer of the OSI model.

Did you notice I said IP-based network? That's really important because by itself, IPsec can't be used to encrypt non-IP traffic. This means that if you run into a situation where you have to encrypt non-IP traffic, you'll need to create a Generic Routing Encapsulation (GRE) tunnel for it (which I explain later) and then use IPsec to encrypt that tunnel!

IPsec Transforms

An *IPsec transform* specifies a single security protocol with its corresponding security algorithm; without these transforms, IPsec wouldn't be able to give us its glory. It's important to be familiar with these technologies, so let me take a second to define the security protocols and briefly introduce the supporting encryption and hashing algorithms that IPsec relies upon.

Security Protocols

The two primary security protocols used by IPsec are *Authentication Header (AH)* and *Encapsulating Security Payload (ESP)*.

Authentication Header (AH)

The AH protocol provides authentication for the data and the IP header of a packet using a one-way hash for packet authentication. It works like this: The sender generates a one-way hash; then the receiver generates the same one-way hash. If the packet has changed in any way, it won't be authenticated and will be dropped. So basically, IPsec relies upon AH to guarantee authenticity. AH checks the entire packet, but it doesn't offer any encryption services.

This is unlike ESP, which only provides an integrity check on the data of a packet.

Encapsulating Security Payload (ESP)

It won't tell you when or how the NASDAQ's gonna bounce up and down like a superball, but ESP will provide confidentiality, data origin authentication, connectionless integrity, anti-replay service, and limited traffic-flow confidentiality by defeating traffic flow analysis—which is almost as good! Anyway, there are five components of ESP:

Confidentiality (encryption) This allows the sending device to encrypt the packets before transmitting in order to prevent eavesdropping. Confidentiality is provided through the use of symmetric encryption algorithms like DES or 3DES. Confidentiality can be selected separately from all other services, but the confidentiality selected must be the same on both endpoints of your VPN.

Data integrity Data integrity allows the receiver to verify that the data received was not altered in any way along the way. IPsec uses checksums as a simple check of the data.

Authentication Authentication ensures that the connection is made with the correct partner. The receiver can authenticate the source of the packet by guaranteeing and certifying the source of the information.

Anti-replay service Anti-replay election is based upon the receiver, meaning the service is effective only if the receiver checks the sequence number. In case you were wondering, a replay attack is when a hacker nicks a copy of an authenticated packet and later transmits it to the intended destination. When the duplicate, authenticated IP packet gets to the destination, it can disrupt services and generally wreak havoc. The *Sequence Number* field is designed to foil this type of attack.

Traffic flow For traffic flow confidentiality to work, you have to have at least tunnel mode selected. It's most effective if it's implemented at a security gateway where tons of traffic amasses because it's precisely the kind of environment that can mask the true source-destination patterns to bad guys who are trying to breach your network's security.

Encryption

VPNs create a private network over a public network infrastructure, but to maintain confidentiality and security, we really need to use IPsec with our VPNs. IPsec uses various types of protocols to perform encryption. The types of encryption algorithms used today are as follows:

Symmetric encryption This encryption requires a shared secret to encrypt and decrypt. Each computer encrypts the data before sending info across the network, with this same key being used to both encrypt and decrypt the data. Examples of symmetric key encryption are Data Encryption Standard (DES), Triple DES (3DES), and Advanced Encryption Standard (AES).

Asymmetric encryption Devices that use asymmetric encryption use different keys for encryption than they do for decryption. These keys are called private and public keys.

Private keys encrypt a hash from the message to create a digital signature, which is then verified via decryption using the public key. Public keys encrypt a symmetric key for secure distribution to the receiving host, which then decrypts that symmetric key using its exclusively held private key. It's not possible to encrypt and decrypt using the same key. This is a variant of public key encryption that uses a combination of both a public and private keys. An example of an asymmetric encryption is Rivest, Shamir, and Adleman (RSA).

As you can see from the amount of information I've thrown at you so far, establishing a VPN connection between two sites takes study, time, and practice. And I am just scratching the surface here! I know it can be difficult at times, and it can take quite a bit of patience. Cisco does have some GUI interfaces to help with this process, and they can be very helpful for configuring VPNs with IPsec. Though highly useful and very interesting, they are just beyond the scope of this book, so I'm not going to delve further into this topic here.

To read about the latest buzz on encryption techniques, take a look at this Cisco URL: `http://www.cisco.com/web/about/security/intelligence/nextgen_crypto.html`.

GRE Tunnels

Generic Routing Encapsulation (GRE) is a tunneling protocol that can encapsulate many protocols inside IP tunnels. Some examples would be routing protocols such as EIGRP and OSPF and the routed protocol IPv6. Figure 7.25 shows the different pieces of a GRE header.

A GRE tunnel interface supports a header for each of the following:

- A passenger protocol or encapsulated protocols like IP or IPv6, which is the protocol being encapsulated by GRE

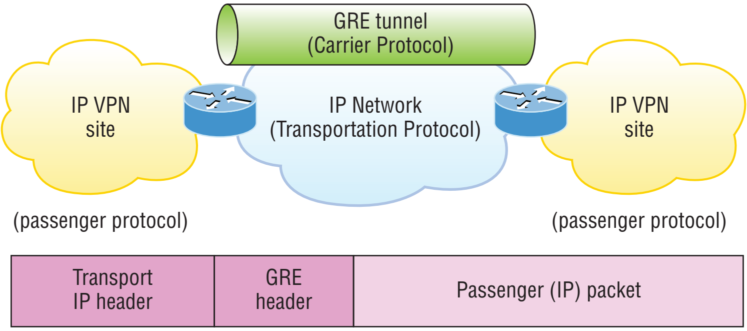
- GRE encapsulation protocol
- A Transport delivery protocol, typically IP

GRE tunnels have the following characteristics:

- GRE uses a protocol-type field in the GRE header so any layer 3 protocol can be used through the tunnel.
- GRE is stateless and has no flow control.
- GRE offers no security.
- GRE creates additional overhead for tunneled packets—at least 24 bytes.

Now let's take a look at how to configure a GRE tunnel. It's actually pretty simple.

FIGURE 7.25 Generic Routing Encapsulation (GRE) tunnel structure


GRE tunnel
(Carrier Protocol)
IP VPN site
IP Network
(Transportation Protocol)
IP VPN site
(passenger protocol)
(passenger protocol)
Transport IP header
GRE header
Passenger (IP) packet

Configuring GRE Tunnels

Before you attempt to configure a GRE tunnel, you need to create an implementation plan. Here's a checklist for what you need to configure and implement a GRE:

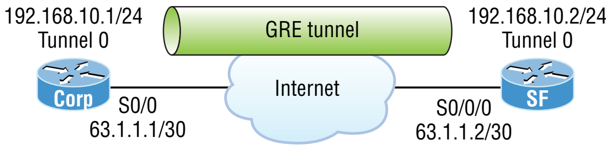
- Use IP addressing.
- Create the logical tunnel interfaces.
- Specify that you're using GRE tunnel mode under the tunnel interface (this is optional since this is the default tunnel mode).
- Specify the tunnel source and destination IP addresses.
- Configure an IP address for the tunnel interface.

Let's take a look at how to bring up a simple GRE tunnel. Figure 7.26 shows the network with two routers.

First, we need to make the logical tunnel with the interface tunnel number command. We can use any number up to 2.14 billion.

```
Corp(config)#int s0/0/0
Corp(config-if)#ip address 63.1.1.1 255.255.255.252
Corp(config)#int tunnel ?
  <0-2147483647>  Tunnel interface number
Corp(config)#int tunnel 0
*Jan 5 16:58:22.719:%LINEPROTO-5-UPDOWN: Line protocol on Interface Tunnel0,
changed state to down
```

FIGURE 7.26 Example of GRE configuration



Once we have configured our interface and created the logical tunnel, we need to configure the mode and then transport protocol.

```
Corp(config-if)#tunnel mode ?
  aurp    AURP TunnelTalk AppleTalk encapsulation
  cayman  Cayman TunnelTalk AppleTalk encapsulation
  dvmrp   DVMRP multicast tunnel
  eon     EON compatible CLNS tunnel
  gre     generic route encapsulation protocol
  ipip    IP over IP encapsulation
  ipsec   IPSec tunnel encapsulation
  iptalk  Apple IPTalk encapsulation
  ipv6    Generic packet tunneling in IPv6
  ipv6ip  IPv6 over IP encapsulation
  nos     IP over IP encapsulation (KA9Q/NOS compatible)
  rbscp   RBSCP in IP tunnel
Corp(config-if)#tunnel mode gre ?
  ip          over IP
  ipv6        over IPv6
  multipoint  over IP (multipoint)


Corp(config-if)#tunnel mode gre ip
```

Okay, now that we've created the tunnel interface, the type, and the transport protocol, we must configure our IP addresses for use inside of the tunnel. Of course, you need to use your actual physical interface IP for the tunnel to send traffic across the Internet, but you also need to configure the tunnel source and tunnel destination addresses.

```
Corp(config-if)#ip address 192.168.10.1 255.255.255.0
Corp(config-if)#tunnel source 63.1.1.1
Corp(config-if)#tunnel destination 63.1.1.2


Corp#sho run interface tunnel 0
Building configuration...


Current configuration : 117 bytes
!
interface Tunnel0
 ip address 192.168.10.1 255.255.255.0
 tunnel source 63.1.1.1
 tunnel destination 63.1.1.2
end
```

Now let's configure the other end of the serial link and watch the tunnel pop up!

```
SF(config)#int s0/0/0
SF(config-if)#ip address 63.1.1.2 255.255.255.252
SF(config-if)#int t0
SF(config-if)#ip address 192.168.10.2 255.255.255.0
SF(config-if)#tunnel source 63.1.1.2
SF(config-if)#tun destination 63.1.1.1
*May 19 22:46:37.099: %LINEPROTO-5-UPDOWN: Line protocol on Interface Tunnel0,
changed state to up
```

Oops—did I forget to set my tunnel mode and transport to GRE and IP on the SF router? No, I didn't need to because it's the default tunnel mode on Cisco IOS. Nice! So, first I set the physical interface IP address (which used a global address even though I didn't have to), then I created the tunnel interface and set the IP address of the tunnel interface. It's really important that you remember to configure the tunnel interface with the actual source and destination IP addresses to use or the tunnel won't come up. In my example, the 63.1.1.2 was the source and 63.1.1.1 was the destination.

Verifying GRP Tunnels

As usual I'll start with my favorite troubleshooting command, `show ip interface brief`.

```
Corp#sh ip int brief
Interface        IP-Address      OK? Method Status                Protocol
FastEthernet0/0  10.10.10.5      YES manual up                    up
Serial0/0        63.1.1.1        YES manual up                    up
FastEthernet0/1  unassigned      YES unset  administratively down down
Serial0/1        unassigned      YES unset  administratively down down
Tunnel0          192.168.10.1    YES manual up                    up
```

In this output, you can see that the tunnel interface is now showing as an interface on my router. You can see the IP address of the tunnel interface, and the Physical and Data Link status show as up/up. So far so good, let's take a look at the interface with the `show interfaces tunnel 0` command.

```
Corp#sh int tun 0
Tunnel0 is up, line protocol is up
  Hardware is Tunnel
  Internet address is 192.168.10.1/24
  MTU 1514 bytes, BW 9 Kbit, DLY 500000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation TUNNEL, loopback not set
  Keepalive not set
  Tunnel source 63.1.1.1, destination 63.1.1.2
  Tunnel protocol/transport GRE/IP
    Key disabled, sequencing disabled
    Checksumming of packets disabled
  Tunnel TTL 255
  Fast tunneling enabled
  Tunnel transmit bandwidth 8000 (kbps)
  Tunnel receive bandwidth 8000 (kbps)
```

The `show interfaces` command shows the configuration settings and the interface status as well as the IP address, tunnel source, and destination address. The output also shows the tunnel protocol, which is GRE/IP. Last, let's take a look at the routing table with the `show ip route` command.

```
Corp#sh ip route
[output cut]
     192.168.10.0/24 is subnetted, 2 subnets
C       192.168.10.0/24 is directly connected, Tunnel0
L       192.168.10.1/32 is directly connected, Tunnel0
     63.0.0.0/30 is subnetted, 2 subnets
C       63.1.1.0 is directly connected, Serial0/0
L       63.1.1.1/32 is directly connected, Serial0/0
```

The `tunnel0` interface shows up as a directly connected interface, and although it's a logical interface, the router treats it as a physical interface, just like serial 0/0 in the routing table.

```
Corp#ping 192.168.10.2
```

```
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.10.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5)
```

Did you notice that I just pinged 192.168.10.2 across the Internet?

Summary

In this chapter, you learned the difference between the following WAN services: cable, DSL, HDLC, PPP, PPPoE, and Frame Relay. You also learned that you can use a VPN once any of those services are up and running, as well as, create and verify a tunnel interface.

It's so important for you to understand High-Level Data-Link Control (HDLC) and how to verify with the `show interface` command that HDLC is enabled! You've been provided with some really important HDLC information as well as information on how the Point-to-Point Protocol (PPP) is used if you need more features than HDLC offers or if you're using two different brands of routers. You now know that this is because various versions of HDLC are proprietary and won't work between two different vendors' routers.

When we went through the section on PPP, I discussed the various LCP options as well as the two types of authentication that can be used: PAP and CHAP.

And we talked about Frame Relay and the two different encapsulation methods used with it in detail. We also discussed LMI options, Frame Relay maps, and subinterface configurations. In addition to the Frame Relay terms and features we covered, I demonstrated Frame Relay configuration and verification in depth.

We finished up the chapter with a discussion on virtual private networks, IPsec, and encryption, and I explained GRE and how to configure the tunnel and then verify it.

Exam Essentials

Remember the default serial encapsulation on Cisco routers. Cisco routers use a proprietary High-Level Data-Link Control (HDLC) encapsulation on all their serial links by default.

Understand the different Frame Relay encapsulations. Cisco uses two different Frame Relay encapsulation methods on its routers: Cisco and IETF. If you are using the Cisco encapsulation method, you are telling your router that a Cisco router is installed on the other side of the PVC. If you are using the IETF encapsulation, you are telling your router that a non-Cisco router is installed on the other side of the PVC.

Remember what the CIR is in Frame Relay. The CIR is the average rate, in bits per second, at which the Frame Relay switch agrees to transfer data.

Remember the commands for verifying and troubleshooting Frame Relay. The `show frame-relay lmi` command will give you the LMI traffic statistics regarding LMI traffic exchanged between the local router and the Frame Relay switch. The `show frame pvc` command will list all configured PVCs and DLCI numbers.

Remember the PPP Data Link layer protocols. The three Data Link layer protocols are Network Control Protocol (NCP), which defines the Network layer protocols; Link Control Protocol (LCP), a method of establishing, configuring, maintaining, and terminating the point-to-point connection; and High-Level Data-Link Control (HDLC), the MAC layer protocol that encapsulates the packets.

Be able to troubleshoot a PPP link. Understand that a PPP link between two routers will show up and a ping would even work between the router if the layer 3 addresses are wrong.

Remember the various types of serial WAN connections. The serial WAN connections that are most widely used are HDLC, PPP, and Frame Relay.

Understand the term *virtual private network*. You need to understand why and how to use a VPN between two sites and the purpose that IPsec serves with VPNs.

Understand how to configure and verify a GRE tunnel. To configure GRE, first configure the logical tunnel with the `interface tunnel` *`number`* command. Configure the mode and transport, if needed, with the `tunnel mode` *`mode protocol`* command, then configure the IP addresses on the tunnel interfaces, the tunnel source and tunnel destination addresses, and your physical interfaces with global addresses. Verify with the `show interface tunnel` command as well as the ping protocol.

Written Lab 7

The answers to this lab can be found in Appendix A, "Answers to Written Labs."

Write the answers to the following WAN questions:

1. Write the command to see the encapsulation method on serial 0/0 of a Cisco router.
2. Write the commands to configure s0/0 to PPP encapsulation.
3. Write the commands to configure a username of *todd* and password of *cisco* that is used on a Cisco router for PPP authentication.
4. Write the commands to enable CHAP authentication on a Cisco serial interface. (Assume PPP is the encapsulation type.)
5. Write the commands to configure the DLCI numbers for two serial interfaces, 0/0 and 0/1. Use 16 for s0/0 and 17 for s0/1.
6. Write the commands to configure a remote office using a point-to-point subinterface. Use DLCI 16 and IP address 172.16.60.1/24.
7. What protocol would you use if you were running xDSL and needed authentication?
8. What are the three protocols specified in PPP?
9. To provide security in your VPN tunnel, what protocol suite would you use?
10. What are the typical three different categories of VPNs?

Hands-on Labs

In this section, you will configure Cisco routers in three different WAN labs using the figure supplied in each lab. (These labs are included for use with real Cisco routers but work perfectly with the LammleSim IOS simulator and with Cisco's Packet Tracer program.)

Lab 7.1: Configuring PPP Encapsulation and Authentication

Lab 7.2: Configuring and Monitoring HDLC

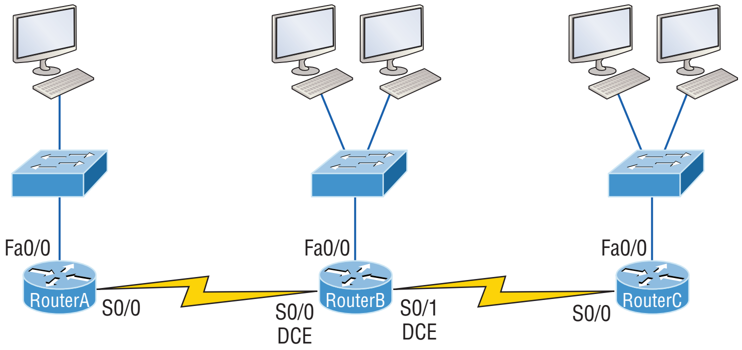
Lab 7.3: Configuring Frame Relay and Subinterfaces

Lab 7.4: Configuring a GRE Tunnel

Hands-on Lab 7.1: Configuring PPP Encapsulation and Authentication

By default, Cisco routers use High-Level Data-Link Control (HDLC) as a point-to-point encapsulation method on serial links. If you are connecting to non-Cisco equipment, then you can use the PPP encapsulation method to communicate.

Labs 7.1 and 7.2 will have you configure the following diagram.



1. Type **`sh int s0/0`** on RouterA and RouterB to see the encapsulation method.
2. Make sure each router has the hostname assigned.

```
RouterA#config t
RouterA(config)#hostname RouterA


RouterB#config t
RouterB(config)#hostname RouterB
```

3. To change the default HDLC encapsulation method to PPP on both routers, use the `encapsulation` command at interface configuration. Both ends of the link must run the same encapsulation method.

```
RouterA#Config t
RouterA(config)#int s0
RouterA(config-if)#encap ppp
```

4. Now go to RouterB and set serial 0/0 to PPP encapsulation.

```
RouterB#config t
RouterB(config)#int s0
RouterB(config-if)#encap ppp
```

5. Verify the configuration by typing **`sh int s0/0`** on both routers.
6. Notice the IPCP and CDPCP (assuming the interface is up). This is the information used to transmit the upper-layer (Network layer) information across the HDLC at the MAC sublayer.

7. Define a username and password on each router. Notice that the username is the name of the remote router. Also, the password must be the same.

```
RouterA#config t
RouterA(config)#username RouterB password todd


RouterB#config t
RouterB(config)#username RouterA password todd
```

8. Enable CHAP or PAP authentication on each interface.

```
RouterA(config)#int s0
RouterA(config-if)#ppp authentication chap


RouterB(config)#int s0
RouterB(config-if)#ppp authentication chap
```

9. Verify the PPP configuration on each router by using these commands.

```
RouterB(config-if)#shut
RouterB(config-if)#debug ppp authentication
RouterB(config-if)#no shut
```

Hands-on Lab 7.2: Configuring and Monitoring HDLC

There really is no configuration required for HDLC (as it is the default configuration on Cisco serial interfaces), but if you completed Lab 7.1, then the PPP encapsulation would be set on both routers. This is why I put the PPP lab first. This lab allows you to actually configure HDLC encapsulation on a router.

NOTE For this second lab, you will use the same configuration you used for Lab 7.1.

1. Set the encapsulation for each serial interface by using the `encapsulation hdlc` command.

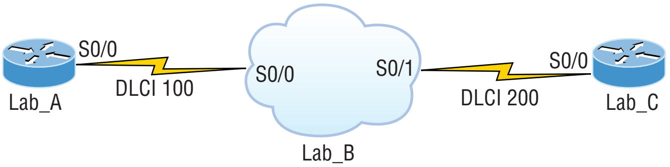
```
RouterA#config t
RouterA(config)#int s0
RouterA(config-if)#encapsulation hdlc
```

```
RouterB#config t
RouterB(config)#int s0
RouterB(config-if)#encapsulation hdlc
```

2. Verify the HDLC encapsulation by using the `show interface s0` command on each router.

Hands-on Lab 7.3: Configuring Frame Relay and Subinterfaces

In this lab, you will use the following diagram to configure Frame Relay, which can easily be created in LammleSim as well as Packet Tracer.



You will configure the Lab_B router to be a Frame Relay switch (this information was not covered in this chapter and is not included in the exam objectives; it will be preconfigured in LammleSim). You will then configure the Lab_A and Lab_C routers to use the switch to bring up the PVC.

1. Set the hostname, using the `frame-relay switching` command, and the encapsulation of each serial interface on the Frame Relay switch.

```
Router#config t
Router(config)#hostname Lab_B
Lab_B(config)#frame-relay switching [makes the router an
FR switch]
Lab_B(config)#int s0/0
Lab_B(config-if)#encapsulation frame-relay
Lab_B(config-if)#int s0/1
Lab_B(config-if)#encapsulation frame-relay
```

2. Configure the Frame Relay mappings on each interface. You do not have to have IP addresses on these interfaces because they are only switching one interface to another with Frame Relay frames.

```
Lab_B(config-if)#int s0/0
Lab_B(config-if)#frame intf-type dce
[The above command makes this an FR DCE interface, which
is different than a router's interface being DCE]
```

```
Lab_B(config-if)#frame-relay route 100 interface
  Serial0/1 200
Lab_B(config-if)#clock rate 64000
[The above command is used if you have this as DCE, which
is different than an FR DCE]
Lab_B(config-if)#int s0/1
Lab_B(config-if)#frame intf-type dce
Lab_B(config-if)#frame-relay route 200 interface
  Serial0/0 100
Lab_B(config-if)#clock rate 64000 [if you have this as DCE]
```

This is not as hard as it looks. The `route` command just says that if you receive frames from PVC 102, send them out `int s0/1` using PVC 201. The second mapping on serial 0/1 is just the opposite. Anything that comes in `int s0/1` is routed out serial0/0 using PVC 102.

3. Configure Lab_A with a point-to-point subinterface.

```
Router#config t
Router(config)#hostname Lab_A
Lab_A(config)#int s0/0
Lab_A(config-if)#encapsulation frame-relay
Lab_A(config-if)#int s0/0.102 point-to-point
Lab_A(config-if)#ip address 172.16.10.1
  255.255.255.0
Lab_A(config-if)#frame-relay interface-dlci 102
```

4. Configure Lab_C with a point-to-point subinterface.

```
Router#config t
Router(config)#hostname Lab_C
Lab_C(config)#int s0/0
Lab_C(config-if)#encapsulation frame-relay
Lab_C(config-if)#int s0/0.201 point-to-point
Lab_C(config-if)#ip address 172.16.10.2
  255.255.255.0
Lab_C(config-if)#frame-relay interface-dlci 201
```

5. Verify your configurations with the following highlighted commands.

```
Lab_A>sho frame ?
 ip       show frame relay IP statistics
 lmi      show frame relay lmi statistics
 map      Frame-Relay map table
```

```
pvc      show frame relay pvc statistics
route    show frame relay route
traffic  Frame-Relay protocol statistics
```

6. Also, use Ping and Telnet to verify connectivity.

Hands-on Lab 7.4: Configuring a GRE Tunnel

In this lab you will configure two point-to-point routers with a simple IP GRE tunnel. You can use a real router, LammleSim, or Packet Tracer to do this lab.

1. First, configure the logical tunnel with the interface tunnel number command.

```
Corp(config)#int s0/0/0
Corp(config-if)#ip address 63.1.1.2 255.255.255.252
Corp(config)#int tunnel ?
  <0-2147483647>  Tunnel interface number
Corp(config)#int tunnel 0
*Jan  5 16:58:22.719: %LINEPROTO-5-UPDOWN: Line protocol
on Interface Tunnel0, changed state to down
```

2. Once you have configured your interface and created the logical tunnel, you need to configure the mode and then the transport protocol.

```
Corp(config-if)#tunnel mode ?
  aurp    AURP TunnelTalk AppleTalk encapsulation
  cayman  Cayman TunnelTalk AppleTalk encapsulation
  dvmrp   DVMRP multicast tunnel
  eon     EON compatible CLNS tunnel
  gre     generic route encapsulation protocol
  ipip    IP over IP encapsulation
  ipsec   IPSec tunnel encapsulation
  iptalk  Apple IPTalk encapsulation
  ipv6    Generic packet tunneling in IPv6
  ipv6ip  IPv6 over IP encapsulation
  nos     IP over IP encapsulation (KA9Q/NOS compatible)
  rbscp   RBSCP in IP tunnel
Corp(config-if)#tunnel mode gre ?
  ip          over IP
  ipv6        over IPv6
  multipoint  over IP (multipoint)


Corp(config-if)#tunnel mode gre ip
```

3. Okay, now that you have created the tunnel interface, the type, and the transport protocol, you need to configure your IP addresses. Of course, you need to use your actual interface IP for the tunnel, but you also need to configure the tunnel source and tunnel destination addresses.

```
Corp(config-if)#int t0
Corp(config-if)#ip address 192.168.10.1 255.255.255.0
Corp(config-if)#tunnel source 63.1.1.1
Corp(config-if)#tunnel destination 63.1.1.2


Corp#sho run interface tunnel 0
Building configuration...


Current configuration : 117 bytes
!
interface Tunnel0
 ip address 192.168.10.1 255.255.255.0
 tunnel source 63.1.1.1
 tunnel destination 63.1.1.2
end
```

4. Now configure the other end of the serial link and watch the tunnel pop up!

```
SF(config)#int s0/0/0
SF(config-if)#ip address 63.1.1.2 255.255.255.252
SF(config-if)#int t0
SF(config-if)#ip address 192.168.10.2 255.255.255.0
SF(config-if)#tunnel source 63.1.1.2
SF(config-if)#tun destination 63.1.1.1
*May 19 22:46:37.099: %LINEPROTO-5-UPDOWN: Line protocol on Interface
Tunnel0, changed state to up
```

Remember, you don't need to configure your tunnel mode and transport protocol because GRE and IP are the defaults. It's really important that you remember to configure the tunnel interface with the actual source and destination IP addresses to use or the tunnel won't come up. In my example, 63.1.1.2 was the source and 63.1.1.1 was the destination.

5. Verify with the following commands:

```
Corp#sh ip int brief
```

You should see that the tunnel interface is now showing as an interface on your router. The IP address of the tunnel interface and the physical and data link status shows as up/up.

```
Corp#sh int tun 0
```

The `show interfaces` command shows the configuration settings and the interface status as well as the IP address and tunnel source and destination address.

```
Corp#sh ip route
```

The tunnel0 interface shows up as a directly connected interface, and although it's a logical interface, the router treats it as a physical interface just like serial0/0 in the routing table.

Review Questions

NOTE The following questions are designed to test your understanding of this chapter's material. For more information on how to get additional questions, please see this book's introduction.

The answers to these questions can be found in Appendix B, "Answers to Chapter Review Questions."

1. Which command will display the CHAP authentication process as it occurs between two routers in the network?

    A. `show chap authentication`
    B. `show interface serial 0`
    C. `debug ppp authentication`
    D. `debug chap authentication`

2. Which command is required for connectivity in a Frame Relay network if Inverse ARP is not operational?

    A. `frame-relay arp`
    B. `frame-relay map`
    C. `frame-relay interface-dci`
    D. `frame-relay lmi-type`

3. Suppose you have a customer who has a central HQ and six branch offices. The customer anticipates adding six more branches in the near future. It wishes to implement a WAN technology that will allow the branches to economically connect to HQ and you have no free ports on the HQ router. Which of the following would you recommend?

    A. PPP
    B. HDLC
    C. Frame Relay
    D. ISDN

4. Which of the following command options are displayed when you use the `Router#`**`show frame-relay ?`** command? (Choose three.)

    A. `dlci`
    B. `neighbors`
    C. `lmi`
    D. `pvc`
    E. `map`

5. How should a router that is being used in a Frame Relay network be configured to keep split horizon issues from preventing routing updates?
   - **A.** Configure a separate subinterface for each PVC with a unique DLCI and subnet assigned to the subinterface.
   - **B.** Combine multiple Frame Relay circuits as a point-to-point line to support multicast and broadcast traffic.
   - **C.** Configure many subinterfaces in the same subnet.
   - **D.** Configure a single subinterface to establish multiple PVC connections to multiple remote router interfaces.

6. Which encapsulations can be configured on a serial interface? (Choose three.)
   - **A.** Ethernet
   - **B.** Token Ring
   - **C.** HDLC
   - **D.** Frame Relay
   - **E.** PPP

7. When setting up Frame Relay for point-to-point subinterfaces, which of the following must not be configured?
   - **A.** The Frame Relay encapsulation on the physical interface
   - **B.** The local DLCI on each subinterface
   - **C.** An IP address on the physical interface
   - **D.** The subinterface type as point-to-point

8. When a router is connected to a Frame Relay WAN link using a serial DTE interface, how is the clock rate determined?
   - **A.** By the CSU/DSU
   - **B.** By the far end router
   - **C.** By the `clock rate` command
   - **D.** By the Physical layer bit stream timing

9. A default Frame Relay WAN is classified as what type of physical network?
   - **A.** Point-to-point
   - **B.** Broadcast multi-access
   - **C.** Nonbroadcast multi-access
   - **D.** Nonbroadcast multipoint

10. Which of the following encapsulates PPP frames in Ethernet frames and uses common PPP features like authentication, encryption, and compression?

    A. PPP

    B. PPPoA

    C. PPPoE

    D. Token Ring

11. You need to configure a router for a Frame Relay connection to a non-Cisco router. Which of the following commands will prepare the WAN interface of the router for this connection?

    A. `Router(config-if)#`**`encapsulation frame-relay q933a`**

    B. `Router(config-if)#`**`encapsulation frame-relay ansi`**

    C. `Router(config-if)#`**`encapsulation frame-relay ietf`**

    D. `Router(config-if)#`**`encapsulation frame-relay cisco`**

12. You have configured a serial interface with GRE IP commands on a corporate router with a point-to-point link to a remote office. What command will show you the IP addresses and tunnel source and destination addresses of the interfaces?

    A. `show int serial 0/0`

    B. `show ip int brief`

    C. `show interface tunnel 0`

    D. `show tunnel ip status`

    E. `debug ip interface tunnel`

13. Which of the following is true regarding WAN technologies? (Choose three.)

    A. You must use PPP on a link connecting two routers using a point-to-point lease line.

    B. You can use a T1 to connect a customer site to the ISP.

    C. You can use a T1 to connect a Frame Relay connection to the ISP.

    D. You can use Ethernet as a WAN service by using EoMPLS.

    E. When using an Ethernet WAN, you must configure the DLCI.

14. You want to allow remote users to send protected packets to the corporate site, but you don't want to install software on the remote client machines. What is the best solution that you could implement?

    A. GRE tunnel

    B. Web VPN

    C. VPN Anywhere

    D. IPsec

15. Why won't the serial link between the Corp router and the Remote router come up?

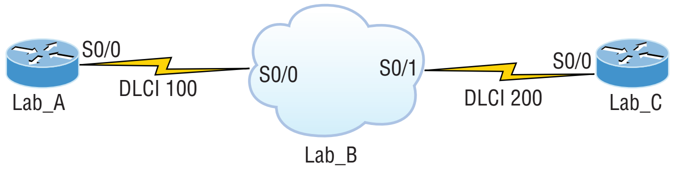
```
Corp#sh int s0/0
Serial0/0 is up, line protocol is down
  Hardware is PowerQUICC Serial
  Internet address is 10.0.1.1/24
  MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec,
     reliability 254/255, txload 1/255, rxload 1/255
  Encapsulation PPP, loopback not set


Remote#sh int s0/0
Serial0/0 is up, line protocol is down
  Hardware is PowerQUICC Serial
  Internet address is 10.0.1.2/24
  MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec,
     reliability 254/255, txload 1/255, rxload 1/255
  Encapsulation HDLC, loopback not set
```

A. The serial cable is faulty.
B. The IP addresses are not in the same subnet.
C. The subnet masks are not correct.
D. The keepalive settings are not correct.
E. The layer 2 frame types are not compatible.

16. Which of the following are benefits of using a VPN in your internetwork? (Choose three.)

A. Security
B. Private high-bandwidth links
C. Cost savings
D. Incompatibility with broadband technologies
E. Scalability

17. A remote site has just been connected to the central office, named Lab_A. However, remote users cannot access applications at the central office. The remote router can be pinged from the Lab_A office router. After reviewing the following command output, which do you think is the most likely reason for the problem?



```
Lab  A#show running-config
!
interface Serial 0/0
 ip address 10.0.8.1 255.255.248.0
 encapsulation frame-relay
 frame-relay map ip 10.0.15.2 200
!
Router rip
Network 10.0.0.0

Lab  C#show running-config
!
interface Serial 0/0
 ip address 10.0.15.2 255.255.248.0
 encapsulation frame-relay
 frame-relay map ip 10.0.8.1 100
!
Router rip
Network 10.0.0.0
```

A. The Frame Relay PVC is down.

B. The IP addressing on the central/remote router link is incorrect.

C. RIP routing information is not being forwarded.

D. Frame Relay Inverse ARP is not properly configured.

18. Which of the following is an industry-wide standard suite of protocols and algorithms that allows for secure data transmission over an IP-based network that functions at the layer 3 Network layer of the OSI model?

A. HDLC

B. Cable

C. VPN

D. IPsec

E. xDSL

19. Which of the following describes the creation of private networks across the Internet, enabling privacy and tunneling of non-TCP/IP protocols?

A. HDLC

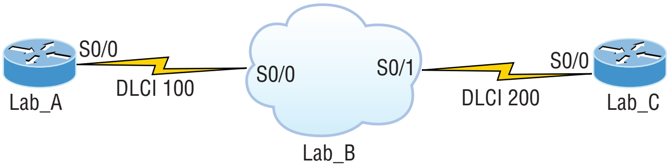
B. Cable

C. VPN

D. IPsec

E. xDSL

20. Referring to the following diagram, what functions does the Frame Relay DLCI provide with respect to router Lab_A?



A. Identifies the signaling standard between Lab_A and the frame switch

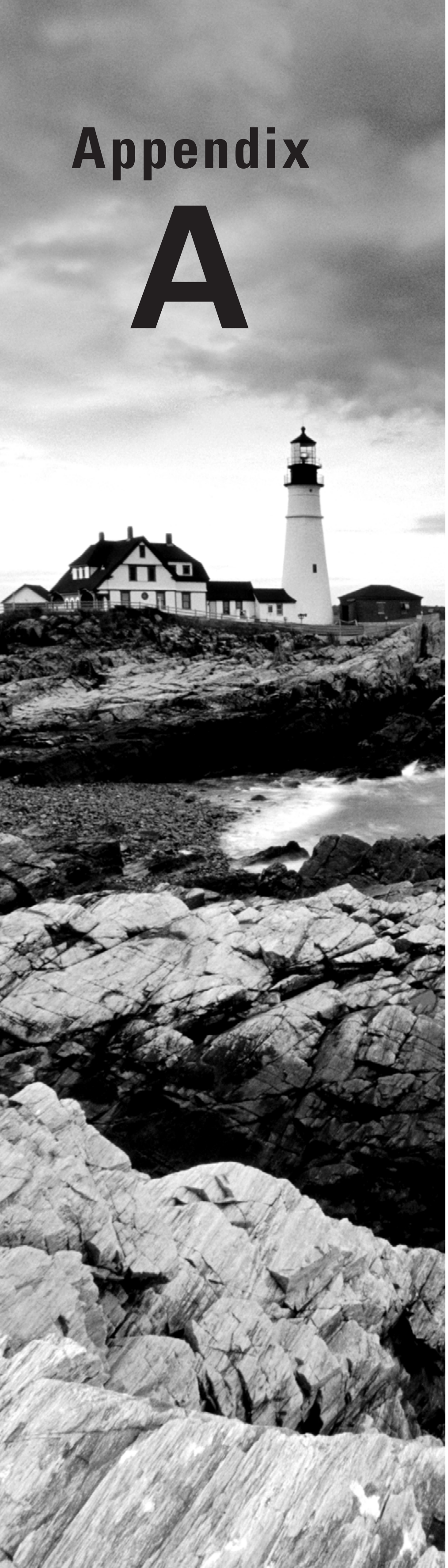
B. Identifies a portion of the virtual circuit between Lab_A and the frame switch

C. Identifies the encapsulation used between Lab_A and Lab_B

D. Defines the signaling standard between Lab_B and the frame switch

Appendix A

Answers to Written Labs



Chapter 1: Enhanced Switch Technologies

1. PAgP
2. `show spanning-tree summary`
3. 802.1w
4. STP
5. BPDU Guard
6. `(config-if)#`**`spanning-tree portfast`**
7. `Switch#`**`show etherchannel port-channel`**
8. `Switch(config)#`**`spanning-tree vlan 3 root primary`**
9. `show spanning-tree`, then follow the root port that connects to the root bridge using CDP, or `show spanning-tree summary`.
10. Active and Passive

Chapter 2: Managing Cisco Devices

Written Lab 2.1: IOS Management

1. `copy flash tftp`
2. 0x2101
3. 0x2102
4. 0x2100
5. UDI
6. 0x2142

7. `boot system`
8. `POST test`
9. `copy tftp flash`
10. `show license`

Written Lab 2.2: Router Memory

1. Flash memory
2. ROM
3. NVRAM
4. ROM
5. RAM
6. RAM
7. ROM
8. ROM
9. RAM
10. RAM

Chapter 3: IP Services

1. `show standby`
2. `show glbp`
3. GLBP
4. NetFlow
5. 5
6. SNMP
7. 4
8. `show ip cache flow`
9. `logging` *`ip_address`*
10. The AVG

Chapter 4: Troubleshooting IP, IPv6, and VLANs

1. The INCMP is an incomplete message, which means a neighbor solicitation message has been sent but the neighbor message has not yet been received.
2. `switchport trunk native vlan 66`
3. Access, auto, desirable, nonegotiate, and trunk (on)
4. Verify that the default gateway is correct. Verify that name resolution settings are correct. Verify that there are no ACLs blocking traffic.
5. `ping ::1`

Chapter 5: Enhanced IGRP

1. `ipv6 router eigrp` *`as`*
2. FF02::A
3. False
4. The routers will not form an adjacency
5. Passive interface

Chapter 6: Multi-Area OSPF

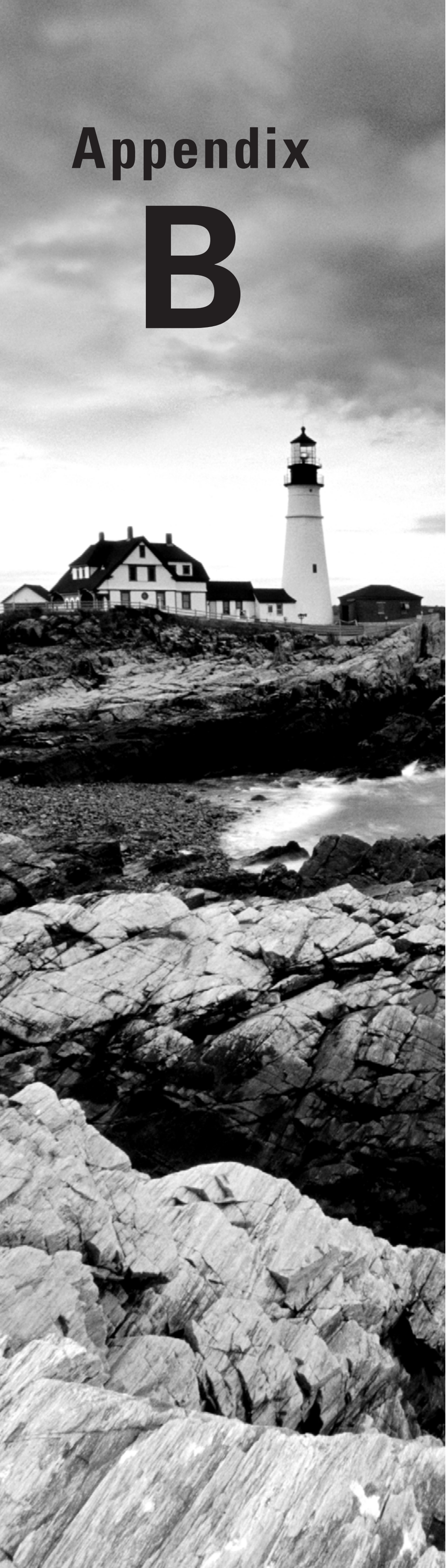
1. Type 5 or type 7
2. 2WAY
3. Type 3, and possibly Type 4 and 5
4. When all LSAs have synchronized with a neighbor on a point-to-point link
5. True
6. EXCHANGE
7. Type 1
8. `ipv6 ospf 1 area 0`
9. OSPFv2 and v3 use the same items when forming an adjacency: hello and dead timers, subnet info, and area ID all must match; authentication must also match if configured.
10. and `show ipv6 protocols`

Chapter 7: Wide Area Networks

1. `sh int s0/0`
2.
```
config t
int s0/0
encap ppp
```
3.
```
config t
username todd password cisco
```
4.
```
config t
int serial0/0
ppp authentication chap
```
5.
```
config t
int s0/0
frame interface-dlci 16
int s0/1
frame interface-dlci 17
```
6.
```
config t
int s0/0
no ip address
encap frame
int s0/0.16 point-to-point
ip address 172.16.60.1 255.255.255.0
frame interface-dlci 16
```
7. PPPoE or PPPoA
8. HDLC, LCP, and NCP
9. IPsec
10. Remote access VPNs, site-to-site VPNs, and extranet VPNs

Appendix B

Answers to Review Questions



Chapter 1: Enhanced Switched Technologies

1. B, D. The switch is not the root bridge for VLAN 1 or the output would tell us exactly that. We can see that the root bridge for VLAN 1 is off of interface G1/2 with a cost of 4, meaning it is directly connected. Use the command `show cdp nei` to find your root bridge at this point. Also, the switch is running RSTP (802.1d), not STP.
2. D. If you have a server or other devices connected into your switch that you're totally sure won't create a switching loop if STP is disabled, you can use something called PortFast on these ports. Using it means the port won't spend the usual 50 seconds to come up while STP is converging. Understand this doesn't completely disable STP, just stops any port configured with PortFast from transitioning the different port states, and immediately puts the port into forwarding state.
3. A, D. It is important that you can find your root bridge, and the `show spanning-tree` command will help you do this. To quickly find out which VLANs your switch is the root bridge for, use the `show spanning-tree summary` command.
4. A. 802.1w is the also called Rapid Spanning Tree Protocol. It is not enabled by default on Cisco switches, but it is a better STP to run because it has all the fixes that the Cisco extensions provide with 802.1d. Remember, Cisco runs RSTP PVST+, not just RSTP.
5. B. The Spanning Tree Protocol is used to stop switching loops in a layer 2 switched network with redundant paths.
6. C. Convergence occurs when all ports on bridges and switches have transitioned to either the forwarding or blocking states. No data is forwarded until convergence is complete. Before data can be forwarded again, all devices must be updated.
7. C, E. There are two types of EtherChannel: Cisco's PAgP and the IEEE's LACP. They are basically the same and there is little difference to configuring them. For PAgP, use

auto or desirable mode, and with LACP use passive or active. These modes decide which method you are using, and they must be configured the same on both sides of the EtherChannel bundle.

8. A, B, F. RSTP helps with convergence issues that plague traditional STP. Rapid PVST+ is based on the 802.1w standard in the same way that PVST+ is based on 802.1d. The operation of Rapid PVST+ is simply a separate instance of 802.1w for each VLAN.

9. D. BPDU Guard is used when a port is configured for PortFast, or it should be used, because if that port receives a BPDU from another switch, BPDU Guard will shut that port down to stop a loop from occurring.

10. C. To allow for the PVST+ to operate, there's a field inserted into the BPDU to accommodate the extended system ID so that PVST+ can have a root bridge configured on a per-STP instance. The extended system ID (VLAN ID) is a 12-bit field, and we can even see what this field is carrying via the `show spanning-tree` command output.

11. C. PortFast and BPDU Guard allow a port to transition to the forwarding state quickly, which is great for a switch port but not for load balancing. You can somewhat load-balance with RSTP, but that is out of the scope of our objectives, and although you can use PPP to configure multilink (bundle links), this is performed on asynchronous or synchronous serial links. Cisco's EtherChannel can bundle up to eight ports between switches.

12. D. If the Spanning Tree Protocol is not running on your switches and you connect them together with redundant links, you will have broadcast storms and multiple frame copies; you will have broadcast storms and multiple frame copies being received by the same destination device.

13. B, C, E. All the ports on both sides of every link must be configured exactly the same or it will not work. Speed, duplex, and allowed VLANs must match.

14. D, F. There are two types of EtherChannel: Cisco's PAgP and the IEEE's LACP. They are basically the same and there is little difference to configuring them. For PAgP, use the auto or desirable mode, and with LACP use the passive or active mode. These modes decide which method you are using, and they must be configured the same on both sides of the EtherChannel bundle.

15. D. You can't answer this question if you don't know who the root bridge is. SC has a bridge priority of 4,096, so that is the root bridge. The cost for SB was 4, with the direct link, but that link went down. If SB goes through SA to SC, the cost would be 4 + 19, or 23. If SB goes to SA to SD to SC, the cost is 4 + 4 + 4 = 12.

16. A, D. To configure EtherChannel, create the port channel from global configuration mode, and then assign the group number on each interface using the active mode to enable LACP, although just configuring the channel-group command under your interfaces will enable the bundle, but options A and D are the best Cisco objective answers.

17. A, D. You can set the priority to any value from 0 through 61,440 in increments of 4,096. Setting it to zero (0) means that the switch will always be a root as long as it has a lower MAC than another switch with its bridge ID also set to 0. You can also force a switch to be a root for a VLAN with the `spanning-tree vlan` *`vlan`* `primary` command.

18. A. By using per-VLAN spanning tree, the root bridge can be placed in the center of where all the resources are for a particular VLAN, which enables optimal path determination.

19. A, C, D, E. Each 802.1d port transitions through blocking, listening, learning, and finally forwarding after 50 seconds, by default. RSTP uses discarding, learning, and forwarding only.

20. A, C, D, E, F. The roles a switch port can play in STP are root, non-root, designated, non-designated, forwarding, and blocking. Discarding is used in RSTP, and disabled could be a role, but it's not listed as a possible answer.

Chapter 2: Managing Cisco Devices

1. B. The default configuration setting is 0x2102, which tells the router to load the IOS from flash and the configuration from NVRAM. 0x2142 tells the router to bypass the configuration in NVRAM so that you can perform password recovery.

2. E. To copy the IOS to a backup host, which is stored in flash memory by default, use the `copy flash tftp` command.

3. B. To install a new license on an ISR2 router, use the `license install` *`url`* command.

4. C. The configuration register provides the boot commands, and 0x2101 tells the router to boot the mini-IOS, if found, and not to load a file from flash memory. Many newer routers do not have a mini-IOS, so as an alternative, the router would end up in ROM monitor mode if the mini-IOS is not found. However, option C is the best answer for this question.

5. B. The `show flash` command will provide you with the current IOS name and size and the size of flash memory.

6. C. Before you start to configure the router, you should erase the NVRAM with the `erase startup-config` command and then reload the router using the `reload` command.

7. D. The command copy `tftp flash` will allow you to copy a new IOS into flash memory on your router.

8. C. The best answer is `show version`, which shows you the IOS file running currently on your router. The `show flash` command shows you the contents of flash memory, not which file is running.

9. C. All Cisco routers have a default configuration register setting of 0x2102, which tells the router to load the IOS from flash memory and the configuration from NVRAM.

10. C. If you save a configuration and reload the router and it comes up either in setup mode or as a blank configuration, chances are the configuration register setting is incorrect.

11. D. The `License boot module` command installs a Right-To-Use license feature on a router.

12. A. The `show license` command determines the licenses that are active on your system. It also displays a group of lines for each feature in the currently running IOS image along with several status variables related to software activation and licensing, both licensed and unlicensed features.

13. B. The `show license feature` command allows you to view the technology package licenses and feature licenses that are supported on your router along with several status variables related to software activation and licensing, both licensed and unlicensed features.

14. C. The `show license udi` command displays the unique device identifier (UDI) of the router, which comprises the product ID (PID) and serial number of the router.

15. D. The `show version` command displays various information about the current IOS version, including the licensing details at the end of the command's output.

16. C. The `license save flash` command allows you to back up your license to flash memory.

17. C. The `show version` command provides you with the current configuration register setting.

18. C, D. The two steps to remove a license are to first disable the technology package and then clear the license.

19. B, D, E. Before you back up an IOS image to a laptop directly connected to a router's Ethernet port, make sure that the TFTP server software is running on your laptop, that the Ethernet cable is a "crossover," and that the laptop is in the same subnet as the router's Ethernet port, and then you can use the `copy flash tftp` command from your laptop.

20. C. The default configuration setting of 0x2102 tells the router to look in NVRAM for the boot sequence.

Chapter 3: IP Services

1. B. You can create and then enter the ACL into the `snmp-server` command when setting your SNMP configuration on a router.

2. C. GLBP is a Cisco proprietary FHRP that can provide true load-balancing. Instead of having active and passive routers, all routers on a LAN can be forwarders with GLBP.

3. C. The default priority is 100, and if you change the priority to a higher setting, you provide the router with a better chance of being active, but only if the FHRP configured routers come up at the same time.

4. A. The command `service sequence-numbers` adds a sequence number to each system message.

5. C. With NetFlow, you can collect information about the flow of traffic on the network, including address and protocols in use. A recent version of NetFlow is version 9, referred to as Flexible NetFlow.

6. F. There are eight different trap levels. If you choose level 4, level 0 through 4 will be displayed, giving you emergency, alert, critical, and error messages, as well as warning messages. Here are the levels and what they perform:

```
Router (config) #logging trap ?
  <0-7>          Logging severity level
  alerts         Immediate action needed               (severity=1)
  critical       Critical conditions                   (severity=2)
  debugging      Debugging messages                    (severity=7)
  emergencies    System is unusable                    (severity=0)
  errors         Error conditions                      (severity=3)
  informational  Informational messages                (severity=6)
  notifications  Normal but significant conditions     (severity=5)
  warnings       Warning conditions                    (severity=4)
```

7. C. HSRP can keep track of outside interfaces to switch routers from passive to active if the WAN on one router goes down, for example.

8. B. The `show ip cache flow` command provides a summary of the NetFlow statistics, including which protocols are in use.

9. B, C, D. NetFlow, SNMP, and syslog are all different applications that provide different features for administrating your network. NetFlow is a very powerful Cisco tool that allows you to see who is doing what, understand network capacity, and even gather accounting information.

10. D. There are eight different trap levels. If you choose level 5, level 0 through 5 will be displayed, giving you emergency, alert, critical, error, and warning messages, as well as notification messages. Here are the levels and what they perform:

```
Router (config) #logging trap  ?
  <0-7>          Logging severity level
  alerts         Immediate action needed              (severity=1)
  critical       Critical conditions                  (severity=2)
  debugging      Debugging messages                   (severity=7)
  emergencies    System is unusable                   (severity=0)
  errors         Error conditions                     (severity=3)
  informational  Informational messages               (severity=6)
  notifications  Normal but significant conditions    (severity=5)
  warnings       Warning conditions                   (severity=4)
```

11. B, C. There is a logging host command configured to 172.16.10.1, so all messages will be sent there, but don't forget that all messages are always sent to the memory buffers on a router as well. Debugging is trap level 7.

12. B, D. You need to have a Cisco router to run the proprietary NetFlow application, plus you need a server to send the information to, and this is called a collector.

13. B. There are eight different trap levels. If you choose level 3, level 0 through 3 will be displayed, giving you emergency, alert, and critical messages, as well as error messages. Here are the levels and what they perform:

```
Router (config) #logging trap  ?
  <0-7>          Logging severity level
  alerts         Immediate action needed              (severity=1)
  critical       Critical conditions                  (severity=2)
  debugging      Debugging messages                   (severity=7)
  emergencies    System is unusable                   (severity=0)
  errors         Error conditions                     (severity=3)
  informational  Informational messages               (severity=6)
  notifications  Normal but significant conditions    (severity=5)
  warnings       Warning conditions                   (severity=4)
```

14. C. The active virtual router (AVR) responds with the virtual MAC addresses to the clients.

15. B. With HSRP, the command is `show standby`; with GLBP it is simply `show glbp`, or even `show glbp brief`.

16. C. There are eight different trap levels. If you choose level 7, level 0 through 7 will be displayed, giving you emergency, alert, critical, error, warning, notification, and informational messages as well as debugging messages. Here are the levels and what they perform:

```
Router (config) #logging trap  ?
  <0-7>          Logging severity level
  alerts         Immediate action needed              (severity=1)
  critical       Critical conditions                  (severity=2)
  debugging      Debugging messages                   (severity=7)
  emergencies    System is unusable                   (severity=0)
  errors         Error conditions                     (severity=3)
  informational  Informational messages               (severity=6)
  notifications  Normal but significant conditions    (severity=5)
  warnings       Warning conditions                   (severity=4)
```

17. C. HSRP is not true load balancing, but it does provide more than one active router in only one situation; that is, if you have your router interface connected to a trunk link and then configured with subinterfaces, no different than router on a stick. However, only one router can be active for a LAN, unlike with GLBP.

18. G. There are eight different trap levels. If you choose level 6, level 0 through 6 will be displayed, giving you emergency, alert, critical, error, warning, and notification messages as well as informational messages. Here are the levels and what they perform:

```
Router (config) #logging trap  ?
  <0-7>          Logging severity level
  alerts         Immediate action needed              (severity=1)
  critical       Critical conditions                  (severity=2)
  debugging      Debugging messages                   (severity=7)
  emergencies    System is unusable                   (severity=0)
  errors         Error conditions                     (severity=3)
  informational  Informational messages               (severity=6)
  notifications  Normal but significant conditions    (severity=5)
  warnings       Warning conditions                   (severity=4)
```

19. C. SNMPv3 supports strong authentication with MD5 or SHA, providing confidentiality (encryption) and data integrity of messages via DES or DES-256 encryption between agents and managers.

20. C, D. Both HSRP and GLBP are Cisco proprietary FHRPs, but only GLBP can load-balance between multiple routers on the same LAN.

Chapter 4: Troubleshooting IP, IPv6, and VLANs

1. D. Positive confirmation has been received confirming that the path to the neighbor is functioning correctly. REACH is good!
2. B. The most common cause of interface errors is a mismatched duplex mode between two ends of an Ethernet link. If they have mismatched duplex settings, you'll receive a legion of errors, which cause nasty slow performance issues, intermittent connectivity, and massive collisions—even total loss of communication!
3. D. You can verify the DTP status of an interface with the `sh dtp interface` *`interface`* command.
4. A. No DTP frames are generated from the interface. Nonegotiate can be used only if the neighbor interface is manually set as trunk or access.
5. D. The command `show ipv6 neighbors` provides the ARP cache on a router.
6. B. The state is STALE when the interface has not communicated within the neighbor reachable time frame. The next time the neighbor communicates, the state will change back to REACH.
7. B. There is no IPv6 default gateway, which will be the link-local address of the router interface, sent to the host as a router advertisement. Until this host receives the router address, the host will communicate with IPv6 only on the local subnet.
8. D. This host is using IPv4 to communicate on the network, and without an IPv6 global address, the host will be able to communicate to only remote networks with IPv4. The IPv4 address and default gateway are not configured into the same subnet.
9. B, C. The commands `show interface trunk` and `show interface` *`interface`* `switchport` will show you statistics of ports, which includes native VLAN information.
10. A. Most Cisco switches ship with a default port mode of auto, meaning that they will automatically trunk if they connect to a port that is on or desirable. Remember that not all switches are shipped as mode auto, but many are, and you need to set one side, to either on or desirable, in order to trunk between switches.

Chapter 5: Enhanced IGRP

1. B. Only the EIGRP routes will be placed in the routing table because it has the lowest administrative distance (AD), and that is always used before metrics.

2. A, C. EIGRP maintains three tables in RAM: neighbor, topology, and routing. The neighbor and topology tables are built and maintained with the use of Hello and update packets.

3. B. EIGRP does use reported distance, or advertised distance (AD), to tell neighbor routers the cost to get to a remote network. This router will send the FD to the neighbor router and the neighbor router will add the cost to get to this router plus the AD to find the true FD.

4. E. Successor routes are going to be in the routing table since they are the best path to a remote network. However, the topology table has a link to each and every network, so the best answer is topology table and routing table. Any secondary route to a remote network is considered a feasible successor, and those routes are found only in the topology table and used as backup routes in case of primary route failure.

5. C. Any secondary route to a remote network is considered a feasible successor, and those routes are found only in the topology table and used as backup routes in case of primary route failure. You can see the topology table with the `show ip eigrp topology` command.

6. B, C, E. EIGRP and EIGRPv6 routers can use the same RID, unlike OSPF, and this can be set with the `eigrp router-id` command. Also a `variance` can be set to provide unequal-cost load balancing, along with the `maximum-paths` command to set the amount of load-balanced paths.

7. C. There were two successor routes, so by default, EIGRP was load-balancing out s0/0 and s0/1. When s0/0 goes down, EIGRP will just keep forwarding traffic out the second link s0/1. s0/0 will be removed from the routing table.

8. D. To enable EIGRPv6 on a router interface, use the command `ipv6 eigrp` as on individual interfaces that will be part of the EIGRPv6 process.

9. C. The path to network 10.10.50.0 out serial0/0 is more than two times the current FD, so I used a `variance` 3 command to load-balance unequal-cost links three times the FD.

10. B, C. First, a maximum hop count of 16 only is associated with RIP, and EIGRP never broadcasts, so we can eliminate A and D as options. Feasible successors are backup routes and stored in the topology table, so that is correct, and if no feasible successor is located, the EIGRP will flood its neighbors asking for a new path to network 10.10.10.0.

11. D. The `show ip eigrp neighbors` command allows you to check the IP addresses, as well as the retransmit interval and queue counts for the neighbors that have established an adjacency.

12. C, E. For EIGRP to form an adjacency with a neighbor, the AS numbers must match, and the metric K-values must match as well. Also, option F could cause the problem; we can see if it is causing a problem from the output given.

13. A, D. Successor routes are the routes picked from the topology table as the best route to a remote network, so these are the routes that IP uses in the routing table to forward traffic to a remote destination. The topology table contains any route that is not as good as the successor route and is considered a feasible successor, or backup route. Remember that all routes are in the topology table, even successor routes.

14. A, B. Option A will work because the router will change the network statement to 10.0.0.0 since EIGRP uses classful addresses by default. Therefore, it isn't technically a wrong answer, but please understand why it is correct for this question. The 10.255.255.64/27 subnet address can be configured with wildcards just as we use with OSPF and ACLs. The /27 is a block of 32, so the wildcard in the fourth octet will be 31. The wildcard of 0.0.0.0 is wrong because this is a network address, not a host address, and the 0.0.0.15 is wrong because that is only a block of 16 and would only work if the mask was a /28.

15. C. To troubleshoot adjacencies, you need to check the AS numbers, the K-values, networks, passive interfaces, and ACLs.

16. C. EIGRP and EIGRPv6 will load-balance across 4 equal cost-paths by default but can be configured to load-balance across equal- and unequal-cost paths, up to 32 with IOS 15.0 code.

17. B, E. EIGRP must be enabled with an AS number from global configuration mode with the `ipv6 router eigrp` *`as`* command if you need to set the RID or other global parameters. Instead of configuring EIGRP with the network command as with EIGRP, EIGRPv6 is configured on a per-interface basis with the `ipv6 eigrp` *`as`* command.

18. C. There isn't a lot to go on from with the output, but that might make this easier than if there were a whole page of output. Since s0/0/2 has lowest FD and AD, that would become the successor route. For a route to become a feasible successor, its reported distance must be lower than the feasible distance of the current successor route, so C is our best answer based on what we can see.

19. C. The network in the diagram is considered a discontiguous network because you have one classful address subnetted and separated by another classful address. Only RIPv2, OSPF, and EIGRP can work with discontiguous networks, but RIPv2 and EIGRP won't work by default (except for routers running the new 15.0 code). You must use the `no auto-summary` command under the routing protocol configuration. There is a passive interface on RouterB, but this is not on an interface between RouterA and RouterB and won't stop an adjacency.

20. A, B, C, D. Here are the documented steps that Cisco says to check when you have an adjacency issue:
    - Interfaces between the devices are down.
    - The two routers have mismatching EIGRP autonomous system numbers.
    - Proper interfaces are not enabled for the EIGRP process.
    - An interface is configured as passive.
    - K values are mismatched.
    - EIGRP authentication is misconfigured.

Chapter 6: Multi-Area OSPF

1. A, B, D. As the size of a single-area OSPF network grows, so does the size of the routing table and OSPF database that have to be maintained. Also, if there is a change in network topology, the OSPF algorithm has to be rerun for the entire network.

2. B. An autonomous system boundary router (ASBR) is any OSPF router that is connected to an external routing process (another AS). An ABR, on the other hand, connects one (or more) OSPF areas together to area 0.

3. B, D, E. In order for two OSPF routers to create an adjacency, the hello and dead timers must match, and they must both be configured into the same area, as well as being in the same subnet. Also, if authentication is configured, that info must match as well.

4. C. The process starts by sending out Hello packets. Every listening router will then add the originating router to the neighbor database. The responding routers will reply with all of their Hello information so that the originating router can add them to its own neighbor table. At this point, we will have reached the 2WAY state—only certain routers will advance beyond this to establish adjacencies.

5. D. If you have multiple links to the same network, you can change the default cost of a link so OSPF will prefer that link over another with the `ip ospf cost` *`cost`* command.

6. B. In the FULL state, all LSA information is synchronized among adjacent neighbors. OSPF routing can begin only after the FULL state has been reached. The FULL state occurs after the LOADING state finishes.

7. B, D, E. Configuring OSPFv3 is pretty simple, as long as you know what interfaces you are using on your router. There are no network statements, OSPFv3 is configured on a per-interface basis. OSPFv2 and OSPFv3 both use a 32-bit RID, and if you have an IPv4 address configured on at least one interface, you do not need to manually set a RID when configuring EIGRPv3.

8. A. Cisco uses a simple calculation of 10^8/bandwidth of a link to determine the cost. This means anything that is 100 Mbps or higher will have a default cost of 1.
9. B. Referred to as a network link advertisement (NLA), Type 2 LSAs are generated by designated routers (DRs). Remember that a designated router is elected to represent other routers in its network, and it establishes adjacencies with them. The DR uses a Type 2 LSA to send out information about the state of other routers that are part of the same network.
10. C. Referred to as summary link advertisements (SLAs), Type 3 LSAs are generated by area border routers. These ABRs send Type 3 LSAs toward the area external to the one where they were generated. The Type 3 LSA advertises networks, and these LSAs advertise inter-area routes to the backbone area (area 0).
11. D. To see all LSAs a router has learned from its neighbors, you need to see the OSPF LSDB, and you can see this with the `show ip ospf database` command.
12. B. Based on the information in the question, the cost from R1 to R2 is 4, the cost from R2 to R3 is 15, and the cost from R3 to R5 is 4. 15 + 4 + 4 = 23. Pretty simple.
13. B, D. Since R3 is connected to area 1 and R1 is connected to area 2 and area 0, the routes advertised from R3 would show as `OI`, or inter-area routes.
14. A, D, E, F. For two OSPF routers to form an adjacency, they must be in the same area, must be in the same subnet, and must have authentication information match, if configured. You need to also check if an ACL is set and if a passive interface is configured.
15. C. The IOS command `show ip ospf neighbor` shows neighbor router information, such as neighbor ID and the state of adjacency with the neighboring router.
16. D. The command `show ip ospf interface` on a default broadcast multi-access network will show you DRs and BDRs on that network.
17. A, C, D, F. It’s hard to tell from this single output what is causing the problem with the adjacency, but we need to check the ACL 10 to see what that is doing, verify that the routers are in the same area and in the same subnet, and see if that passive interface is configured with the interface we’re using.
18. B, D, G. The default reference bandwidth is 100 by default, and you can change it under the OSPF process with the `auto-cost reference bandwidth` *`number`* command, but if you do, you need to configure this command on all routers in your AS.
19. A, D. Anything Fast Ethernet or above would have a default cost of 1, and serial interfaces have a default cost of 64.
20. D. Anything Fast Ethernet or above would have a default cost of 1, and serial interfaces have a default cost of 64.

Chapter 7: Wide Area Networks

1. C. The command `debug ppp authentication` will show you the authentication process that PPP uses across point-to-point connections.
2. B. If you have a router in your Frame Relay network that does not support IARP, you must create Frame Relay maps on your router, which provide known DLCI-to-IP-address mappings.
3. C. The key is "there are no free ports" on your router. Only Frame Relay can provide a connection to multiple locations with one interface, and in an economical manner no less.
4. C, D, E. The `show frame-relay ?` command provides many options, but the options available in this question are `lmi`, `pvc`, and `map`.
5. A. If you have a serial port configured with multiple DLCIs connected to multiple remote sites, split horizon rules (discussed in Chapter 5) stop route updates received on an interface from being sent out the same interface. By creating subinterfaces for each PVC, you can avoid the split horizon issues when using Frame Relay.
6. C, D, E. Ethernet and Token Ring are LAN technologies and cannot be configured on a serial interface. PPP, HDLC, and Frame Relay are layer 2 WAN technologies that are typically configured on a serial interface.
7. C. It is very important to remember when studying the CCNA R/S exam objectives and when configuring Frame Relay with point-to-point subinterfaces, that you do not put an IP address on the physical interface.
8. A. Clocking on a serial interface is always provided by the CSU/DSU (DCE device). However, if you do not have a CSU/DSU in your nonproduction test environment, then you need to supply clocking with the `clock rate` command on the serial interface of the router with the DCE-cable attached.
9. C. Frame Relay, by default, is a nonbroadcast multi-access (NBMA) network, which means that broadcasts, such as RIP updates, will not be forwarded across the link by default.
10. C. PPPoE encapsulates PPP frames in Ethernet frames and uses common PPP features like authentication, encryption, and compression. PPPoA is used for ATM.
11. C. If you have a Cisco router on one side of a Frame Relay network and a non-Cisco router on the other side, you would need to use the Frame Relay encapsulation type of IETF. The default is Cisco encapsulation, which means that a Cisco router must be on both sides of the Frame Relay PVC.
12. C. The `show interfaces` command shows the configuration settings and the interface status as well as the IP address, and tunnel source and destination address.

13. B, C, D. This is just a basic WAN question to test your understanding of connections. PPP does not need to be used, so option A is not valid. You can use any type of connection to connect to a customer site, so option B is a valid answer. You can also use any type of connection to get to the Frame Relay switch, as long as the ISP supports it, and T1 is valid, so option C is okay. Ethernet as a WAN can be used with Ethernet over MPLS (EoMPLS); however, you don't need to configure a DLCI unless you're using Frame Relay, so E is not a valid answer for this question.

14. B. All web browsers support Secure Sockets Layer (SSL), and SSL VPNs are known as Web VPNs. Remote users can use their browser to create an encrypted connection and they don't need to install any software. GRE doesn't encrypt the data.

15. E. This is an easy question because the Remote router is using the default HDLC serial encapsulation and the Corp router is using the PPP serial encapsulation. You should go to the Remote router and set that encapsulation to PPP or change the Corp router back to the default of HDLC by typing **`no encapsulation`** under the interface.

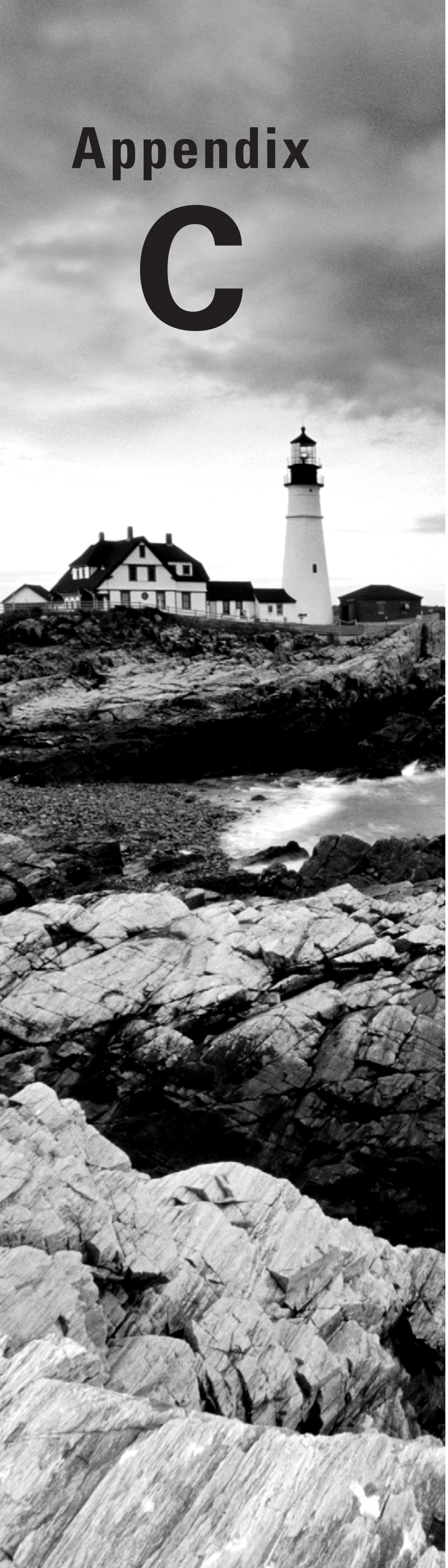
16. A, C, E. VPNs can provide very good security by using advanced encryption and authentication protocols, which will help protect your network from unauthorized access. By connecting the corporate remote offices to their closest Internet provider and then creating a VPN tunnel with encryption and authentication, you'll gain a huge savings over opting for traditional leased point-to-point lines. VPNs scale very well to quickly bring up new offices or have mobile users connect securely while traveling or when connecting from home. VPNs are very compatible with broadband technologies.

17. C. Even though the IP addresses don't look correct, they are in the same subnet, so option B is not correct. The question states that you can ping the other side, so the PVC must be up—option A can't be correct. You cannot configure IARP, so only option C can be correct. Since a Frame Relay network is a nonbroadcast multi-access network by default, broadcasts such as RIP updates cannot be sent across the PVC unless you use the broadcast statement at the end of the `frame-relay map` command.

18. D. IPsec is an industry-wide standard suite of protocols and algorithms that allows for secure data transmission over an IP-based network that functions at the layer 3 Network layer of the OSI model.

19. C. A VPN allows or describes the creation of private networks across the Internet, enabling privacy and tunneling of TCP/IP protocols. A VPN can be set up across any type of link.

20. B. As I mentioned many times in this chapter, and you need to remember this, DLCIs are locally significant only and define the circuit from the router to the switch only. They do not reference a remote router or DLCI. Lab_A would use DLCI 100 to get to the Lab_B networks. RouterB would use DLCI 200 to get to the Lab_A networks.



About the Additional Study Tools

IN THIS APPENDIX:

✓ Additional study tools

✓ System requirements

✓ Using the study tools

✓ Troubleshooting

Additional Study Tools

The following sections are arranged by category and summarize the software and other goodies you'll find on the book's companion website. If you need help with installing the items, refer to the installation instructions in the "Using the Study Tools" section of this appendix.

> **NOTE** The additional study tools can be found at `www.sybex.com/go/ccnaicnd2`. Here, you will get instructions on how to download the files to your hard drive.

Test Engine

The files contain a test engine, which includes practice test questions, as well as the preassessment questions and the chapter review questions, which are also included in the book itself.

Electronic Flashcards

These handy electronic flashcards are just what they sound like. One side contains a question, and the other side shows the answer.

Videos

Author Todd Lammle has provided approximately 30 minutes of companion videos to walk through some of the more difficult concepts of the book.

Network Simulator

Readers can get access to a network simulator that can be used to perform all of the hands-on labs in the book.

PDF of Glossary of Terms

We have included an electronic version of the Glossary in `.pdf` format. You can view the electronic version of the Glossary with Adobe Reader.

Adobe Reader

We've also included a link to download the latest version of Adobe Reader. For more information on Adobe Reader or to check for a newer version, visit Adobe's website at `www.adobe.com/products/reader/`.

System Requirements

Make sure your computer meets the minimum system requirements shown in the following list. If your computer doesn't match up to most of these requirements, you may have problems using the software and files. For the latest and greatest information, please refer to the ReadMe file located in the downloads.

- A PC running Windows XP or newer
- An Internet connection

Using the Study Tools

To install the items, follow these steps:

1. Download the `.ZIP` file to your hard drive, and unzip to an appropriate location. Instructions on where to download this file can be found here: `www.sybex.com/go/ccentsg`.
2. Click the `Start.EXE` file to open up the study tools file.
3. Read the license agreement, and then click the Accept button if you want to use the study tools.

The main interface appears. The interface allows you to access the content with just one or two clicks.

Troubleshooting

Wiley has attempted to provide programs that work on most computers with the minimum system requirements. Alas, your computer may differ, and some programs may not work properly for some reason.

The two likeliest problems are that you don't have enough memory (RAM) for the programs you want to use or you have other programs running that are affecting installation or running of a program. If you get an error message such as "Not enough memory" or "Setup cannot continue," try one or more of the following suggestions and then try using the software again:

Turn off any antivirus software running on your computer. Installation programs sometimes mimic virus activity and may make your computer incorrectly believe that it's being infected by a virus.

Close all running programs. The more programs you have running, the less memory is available to other programs. Installation programs typically update files and programs; so if you keep other programs running, installation may not work properly.

Have your local computer store add more RAM to your computer. This is, admittedly, a drastic and somewhat expensive step. However, adding more memory can really help the speed of your computer and allow more programs to run at the same time.

Customer Care

If you have trouble with the book's companion study tools, please call the Wiley Product Technical Support phone number at (800) 762-2974, or contact them at `http://sybex.custhelp.com/`.

Index

Note to the reader: Throughout this index boldfaced page numbers indicate primary discussions of a topic. *Italicized* page numbers indicate illustrations.

A

AAL5. *See* ATM Adaptation Layer 5
ABRs. *See* area border routers
access control list (ACL), 363, 371
  EIGRP troubleshooting, 213
  OSPF troubleshooting, 257
  SNMP configuration, 116
  troubleshooting, 130, 138–139, 356
access rate, 311
ACK, 172
ACK packet, 177
ACL. *See* access control list
active configuration, 212
active router, 90, *92*, 94, *95*
active timer, 94
active virtual forwarder (AVF), 103
active virtual gateway (AVG), 103, 355
active virtual router (AVR), 365
AD. *See* administrative distance; advertised distance
address resolution states, 145–146
address-family ipv4 autonomous-system, 181
adjacencies, 209, 210
  multi-area OSPF network requirements, **236–237**, 369, 370
  OSPF troubleshooting, 254
  OSPFv3, 263, 265
  troubleshooting, 368, 369
  verifying, 254
administrative distance (AD), 367
advanced distance-vector protocols, 170, 171
Advanced Encryption Standard (AES), 332
advertised distance (AD), 172, 173, 367
AES. *See* Advanced Encryption Standard
AH. *See* Authentication Header
alternate port, *5*
ANSI LMI format, 315, 323
anti-replay service, 331
AppleTalk, 178
area border routers (ABRs), 234, 237–238, 240, 268
ARP cache, 163
ARP requests, 128
AS. *See* autonomous system
AS external link advertisements, 240
ASBRs. *See* autonomous system boundary routers
asymmetric encryption, 332
asymmetrical DSL, 293, **294**
Asynchronous Transfer Mode (ATM), 290, 372
  DSL connections, 294
  PPP over, 294
ATM. *See* Asynchronous Transfer Mode
ATM Adaptation Layer 5 (AAL5), 294
ATTEMPT state, 242
ATU-Cs, 294
authentication, 302
  adjacency requirements, 237
  EIGRP troubleshooting, 204, 208
  Hello protocol, 241
  IPsec, 330, 331
  PPP, *304*, 357, 371
    configuring, **303–304**
    debugging, **306**
    failed, *305*
    methods, **303**
  SNMP, 113
Authentication Header (AH), **331**
auto-cost reference bandwidth, 371
autonegotiation, 135

autonomous system (AS), 172, 203, 217
  EIGRP configuration, 172, 179–181
  external, 238
  external link advertisements, 240
  hop count limits, 171
  mismatched numbers, 204, 208
  OSPFv3, 262
  route advertising, 240
autonomous system boundary routers (ASBRs), 234, 237, 268, 369
  external AS connections, 238
  LSAs, 240
auto-summarization, 181–184, 211
  disabling, 183–184, 211
  discontiguous network confusion, 183, 204
AVF. *See* active virtual forwarder
AVG. *See* active virtual gateway
AVR. *See* active virtual router

B

B2B communications. *See* business-to-business communications
backbone area, 237, 239
backbone routers, 234, 237, 268
backup designated router (BDR), 246
backup port, *5*
Backward Explicit Congestion Notification (BECN), 316, 317, 321
bandwidth, 296
  dynamic allocation, 288
  Frame Relay, 311
  reference, 262, 371
  WAN connection, **286**
BDR. *See* backup designated router
BECN. *See* Backward Explicit Congestion Notification
BGP. *See* Border Gateway Protocol
bit-oriented protocols, 298
bits
  boot field, *52*
  bridge ID, 11, *12*
  classful network address, 179
  configuration register, **51–52**, *52*
    load from NVRAM, *55*
    password recovery, *55*, *56*
  congestion, 316
  EIGRP configuration, 179
  HSRP MAC address, 91, 92
  IPv6 addresses, 141
blocked port, *5*
blocking state, *5–6*, 14, 28, 360
boot, *53–54*
boot field, 52
boot sequence, **49–50**, 74
  interrupting, **56**
boot system, *53–55*, *355*
boot system commands, **53–55**, 61
boot system flash, *54*, 61
boot system rom, *54–55*
boot system tftp, *54–55*
bootloader, 49
bootstrap, 50
Border Gateway Protocol (BGP), 238
bounded updates, 171
BPDU. *See* Bridge Protocol Data Unit
BPDU Guard, **26**, 33, 354, 361
bridge ID, 8, 16
  bits, 11, *12*
  designated port selection, 16
  modifying and verifying, **17–22**
  roles in STP, **4**
  root bridge election, 9
  VLAN troubleshooting, 151
bridge port
  PAgP, 29
  roles, 4–5
Bridge Protocol Data Unit (BPDU), 4, 11
broadcast storms, 2
business-to-business communications (B2B communications), 330
byte-oriented protocols, 298

C

cable, 284, 289, **291**, **292–293**
cable TV (CATV), 289, 292
cabling
  crossover, 58, 78, 363
  demarc to CPE, 285
  faulty, 129, 163
  input errors, 135
  IP network connectivity troubleshooting, 129, 132, 134, 163
  IPv6 network connectivity troubleshooting, 142
  serial WAN, **296–298**
    DTEs and DCEs, **297–298**, 372
    serial transmission, **296**
CAM. *See* content addressable memory
carrier detect (CD), 50
CATV. *See* cable TV
CD. *See* carrier detect
CDP, 160–162
CDPCP. *See* Cisco Discovery Protocol Control Protocol
cellular 3G/4G, 290
central office (CO), 285, 289, 294
Challenge Handshake Authentication Protocol (CHAP), 302, 303, 306
channel groups, 28–32, 33
channel-group, 29–31
CHAP. *See* Challenge Handshake Authentication Protocol
CIR. *See* Committed Information Rate
circuit switching, 286–287
Cisco AnyConnect SSL VPN client, 329
Cisco Discovery Protocol Control Protocol (CDPCP), 301
Cisco encapsulation, 312, 324, 337, 372
Cisco IOS File System (Cisco IFS), **63–68**
  commands, 64–65, 74
  IOS upgrade, **65–68**
Cisco License Manager (CLM), 69
Cisco LMI format, 315, 323
Cisco Long Range Ethernet (LRE), **295–296**
Cisco Prime, 112
Cisco Product License Registration portal, 69
Cisco routers
  boot sequence, **49–50**, 74
  internal components, **48–49**, 74
  PPP configuration on, **303**
Cisco Technical Assistance Center (TAC), 186
classful boundaries, 183
classful network address, 179
classless routing protocols, 171, 181
client redundancy, **86–87**
Clientless Cisco SSL VPN, 329
CLM. *See* Cisco License Manager
clock rate, 297–298
clocking, 297–298, 371
CO. *See* central office
collision domains, 129
Committed Information Rate (CIR), **311–312**, 337
Common Spanning Tree (CST), 9, **10**
community antenna television, 292
config t, 209
config-register, 53, 56
configuration
  active, 212
  EIGRP, **178–197**, 220
  EIGRPv6, 215–217, 220
  GLBP, **104–107**
  GRE tunnels, **333–335**, 338
  HDLC, 298–300, *299*
  HSRP, **95–99**
  LCP, **301–302**
  multi-area networks, **243–245**
  NetFlow, **118–119**
  OSPFv3, 263–265, 268
  port channels, **29–32**
  PPP
    authentication methods, **303–304**
    on Cisco routers, **303**
  SNMP, **114–116**
  subinterfaces, 318–319
  syslog, **109–112**
  viewing and changing, **57**

configuration modes
  configuration register, 53
  EIGRP, 178, 368
  EIGRPv6, 214–215
  global, 25, 29, 53, 361, 368
  interface, 214, 263
  OSPFv3, 263
  port channels, 29
  router, 178, 263
  syslog, 111
configuration register, 49, 74, 362, 363
  changing, **56**
  checking current value, **52–53**
  load from NVRAM, 55
  managing, **50–57**
  password recovery, 55, 56
  resetting, **57**
  understanding bits, **51–52**, *52*
congestion control, 288
  Frame Relay, **316–317**
console line, 107
console port
  restoring IOS, 63
  syslog messages, 107
content addressable memory (CAM), 149
convergence, 361
  EIGRPv6, 214
  STP, **6**, 360
copy flash tftp, 58, 60, 354
copy source-url destination-url, 64
copy startup-config running-config, 57
copy tftp flash, 61, 66, 355
CPE. *See* customer premises equipment
crossover cable, 58, 78, 363
CST. *See* Common Spanning Tree
CSU/DSU, 285, 371
customer premises equipment (CPE), 285, 286, 289
  DSL, 294

D

DAD. *See* duplicate address detection
data communication equipment (DCE), 286, **297–298**, 322
Data Encryption Standard (DES), 331, 332
Data license, 69
Data Link Connection Identifiers (DLCIs), **313–314**
  debug frame lmi information, 323
  Frame Relay maps, 325, 371
  locally significant, 373
  show frame map information, 323
  show frame pvc information, 321–322, 338
  show interface information, 322–323
  subinterface configuration, 319–320
  WAN connections, 372
Data Link layer, 128
Data Over Cable Service Interface Specification (DOCSIS), 292
data terminal equipment (DTE), 286, **297–298**, 312, 322
Database Description packets (DD), 236, 243
database synchronization, 236
DCE. *See* data communication equipment
DD. *See* Database Description packets
DE. *See* Discard Eligibility
dead timer, 356
debug frame lmi, **323–324**
debug ppp authentication, 306, 371
debug standby, 99
DEC. *See* Digital Equipment Corporation
dedicated connections, 286
default gateway, 86, *87*, 356, 367
  ARP request for, 128
  confirming, 131
  FHRPs, 88–90
  HSRP load balancing, 100
  IP network connectivity troubleshooting, 128–133
  IPv6 network connectivity troubleshooting, 141–142, 146–147
  virtual routers, 101
default trap level, 110
delete, 66
demarcation point (demarc), 285, 297
deny, 138
DES. *See* Data Encryption Standard
designated forwarding port, 14, 18

designated port, 4, 15–16
  RSTP, 13
designated routers (DRs), 239, 246, 249, 370
  backup, 246
  NBMA networks, 325
dial-up modems, 286–287, 300
Diffusing Update Algorithm (DUAL), 171, **177**, 219, 220
  EIGRPv6, 214
  path calculation, 174
digital carrier systems, 286
Digital Equipment Corporation (DEC), 2
Digital Signal 0 (DS0), 286
digital subscriber line (DSL), 284, 289, **291**, **293–294**
Digital Subscriber Line Access Multiplexer (DSLAM), 289, 294
Dijkstra algorithm, 236
disabled ports, *5*
Discard Eligibility (DE), 316, 321
discarding state, 13
discontiguous networks, 184, 211, 213, 369
  auto-summarizing confusing, 183, 204
  EIGRP support, 171, 181–183, *182*
  troubleshooting EIGRP, 204, 213
Dish Network, 290
distance-vector routing protocols, 171
distribution network, 292
DLCIs. *See* Data Link Connection Identifiers
DOCSIS. *See* Data Over Cable Service Interface Specification
DOWN state, 241–242
DRs. *See* designated routers
DS0. *See* Digital Signal 0
DS1, 286
DSL. *See* digital subscriber line
DSL interface cards, 294
DSLAM. *See* Digital Subscriber Line Access Multiplexer
DTE. *See* data terminal equipment
DTP. *See* Dynamic Trunking Protocol
DUAL. *See* Diffusing Update Algorithm
duplex settings, 134–135, 366
duplicate address detection (DAD), 141
dynamic auto ports, 158
dynamic bandwidth allocation, 288
dynamic route recoveries, 177
Dynamic Trunking Protocol (DTP), 157–160, 366
  disabling, 158
  frame tagging, 160
  viewing statistics, 157

E

E0, 286
EEPROM. *See* electronically erasable programmable read-only memory
EIA-530, 296
EIA/TIA-232, 296
EIA/TIA-232-C, 300
EIA/TIA-449, 296
802.1d, 2, 6, 9, 11
802.1q, 158, 160–161, 163
802.1w, 2, 5, 10, 360
EIGRP. *See* Enhanced Interior Gateway Protocol
eigrp router-id, 367
EIGRPv6, 170, **214–219**, 367
  configuring, 215–217, 220
  RIDs, 244
  verifying, 217–219
elections
  GLBP preemption, 103
  HSRP positions, 90, 95, 96, 98
  multi-area OSPF, 241–243, 246, 249, 266
  point-to-point links, 246
  root bridge, 8–10
  serial point-to-point links, 266
electronic flashcards, 376
electronically erasable programmable read-only memory (EEPROM), 49
enable-secret setting, 57
encap frame, 357
Encapsulating Security Payload (ESP), **331–332**

encapsulation, 294, 298, 300. *See also* Generic Routing Encapsulation
  ADSL, 294
  Cisco, 312, 324, 337, 372
  EtherChannel, 30
  Ethernet, 288
  Frame Relay, 337
    implementation, 318
    monitoring, 322
    types, **312**
  HDLC, 289, 298–300, 372
  IETF, 312, 324, 337, 372
  layer 2 WAN, 288
  mismatched, **307–308**
  PPP, 300–306
  PPPoA, 294
  PPPoE, 289, 294, 372
  serial, 288, 300, 337, 372
    troubleshooting, 324–325, *325*, *326*
  trunk, 159–162
  tunneling, 330
encapsulation ?, 288
encapsulation frame-relay, 312, 317
encapsulation frame-relay ?, 324
encapsulation frame-relay ietf, 324
encapsulation ppp, 303
encryption
  IPsec, 330, 331
  passwords, 304
  PPPoA, 294
  PPPoE, 289, 372
  SNMPv3, 113, 366
  SSL, 329, 372
  VPNs, 329, **332**, 372
Enhanced Interior Gateway Protocol (EIGRP), 28, 128, **367–369**. *See also* EIGRPv6
  advertisement, 172
  ASs, 172, 179–181
  configuring, **178–197**, 220
    auto-summarization, 181–184
    command modes, 178
    discontiguous networks, 182–183
    hop count, 171, **187–188**
    maximum paths, **187–188**
    metrics, **185–187**
    session initiation, 179
    split horizon, **195–197**
    subnets, 182–185
    summarization, **181–184**
    traffic control, **184–185**
    unequal-cost load balancing, **193–195**
    VLSMs, **181–184**
    wildcards, 180
  default hop count, 171
  DUAL, **177**
  events, 203
  features and operations, **170–178**, 220
  Frame Relay routing problems, 325–327, *326*
  Hellos, 172
  metrics calculation, 171
  multicast traffic, 176
  neighbor discovery, **171–176**
  neighbor table, 220
  OSPF troubleshooting, 258–260
  RIDs, 244
  route discovery and maintenance, **178**
  route selection, **188–193**
    load balancing, 189, 191–192, 368
    routing table, 188–189
    topology table, 189–190, 191
  routing databases, 176
  routing updates, 185
  RTP, **176–177**
  tables used by, 174, 175
  verifying and troubleshooting, **197–214**, 220
    ACLs, 213
    discontiguous networks, 204, 213
    event logs, 203
    K values, 204, 207–210, 213
    load-balancing output, 201–202
    local router status, 199
    mapping network, 198
    passive state routes, 201
    routing protocols, 202–203
    routing table, 200

    topology table, 200–201
    traffic data, 202
    troubleshooting commands, *198*
EoMPLS. *See* Ethernet over MPLS
equal-cost load balancing, 200–201, 368
erase startup-config, 362
error disabled state, 26
ESP. *See* Encapsulating Security Payload
EtherChannel, 24, **28–32**, 33, 360, 361
Ethernet encapsulation, 288
Ethernet over MPLS (EoMPLS), 290, 372
evaluation licenses. *See* Right-To-Use licensing
EXCHANGE state, 243
EXSTART state, 242
extended system ID, 11, 361
extranet VPNs, 330, 357

F

faulty cables, 129, 163
feasible distance (FD), 173, 174, 190, 200, 201, 368
feasible successor (FS), 175, 368
FECN. *See* Forward Explicit Congestion Notification
FHRPs. *See* first hop redundancy protocols
first hop, 88
first hop redundancy protocols (FHRPs), 86, **88–90**, *89*, 120, 363, 366. *See also* Gateway Load Balancing Protocol; Hot Standby Router Protocol; Virtual Router Redundancy Protocol
flash memory, 49, 50, 355
  backing up, 60
  removable, 60, 68
  verifying, **59–60**
  working with, 67
Flexible NetFlow, 116, 364
Forward Explicit Congestion Notification (FECN), 316, 317, 321
forwarding port, 4
forwarding state, 5–6, 13, 360
frame interface-dlci, 357
Frame Relay, 284, 288, **309–328**, 371–373
  ATM, 290
  bandwidth specifications, 311
  CIR, **311–312**, 337
  congestion control, **316–317**
  encapsulation
    implementation, 318
    monitoring, 322
    types, **312**, 337
  IARP, 313
  implementation and monitoring, **317–328**
    single interface, **317–318**
    subinterfaces, **318–320**, 371
  mappings, *324*, 324–325, 371
  monitoring, **320–324**
    show interface, **322–323**
  MPLS replacing, 290
  packet switching, 287
  Physical layer connections, 296
  routing problems, *325*, 325–327, *326*
    EIGRP, *326*, 326–327
    OSPF, 325–326, *326*
    subinterfaces, 327–328
  serial interfaces, 313
  split horizon problems, 197
  troubleshooting, **324–328**, 338
    routing, *325*, 325–327, *326*
    subinterfaces, 327–328
    subnets, 327
  verification, 338
frame tagging
  changing method, 159–160
  switch compatibility, 158
frame-relay interface-dlci, 314, 318
frame-relay interface-dlci *dlci*, 327–328
frame-relay lmi-type, 315
frame-relay lmi-type ansi, 317
frame-relay map ip, 325
frames, 128–129
  HDLC format, *299*
  multiple copies, 2
  PPPoE, 289
  untagged, 159, 163
FS. *See* feasible successor
FULL state, 241–242, 243, 370

G

Gateway Load Balancing Protocol (GLBP), 86, 90, **102–107**, 120, 355, 363, *365–366*
  AVF, 103
  AVG, 103
  configuring, **104–107**
  features, **103–104**
  functions, **103**
  per-host traffic balancing, **104**
gateways
  AVG, 103, *355*
  default, 86, *87*, 356
    ARP request for, 128
    confirming, 131
    FHRPs, 88–90
    HSRP load balancing, 100
    IP network connectivity troubleshooting, 128–133
    IPv6, 366
    IPv6 network connectivity troubleshooting, 141–142, 146–147
    virtual routers, 101
  redundancy
    GLBP, 90
    HSRP, 89–90
    VRRP, 90
  security, 332
  virtual, 103
Generic Routing Encapsulation (GRE), 284, 330
  headers, 332–333
  tunnels, **332–337**, 338
    configuring, **333–335**, 338
    structure, *333*
    verifying, **335–337**, 338
GET BULK, 113
GET messages, 112
glbp, 105–106
GLBP. *See* Gateway Load Balancing Protocol
global addresses, 140, 366
  LMI, 314
global configuration mode, 25, 29, 53, 361, 368
GRE. *See* Generic Routing Encapsulation
groups
  channel, 28–32, 33
  GLBP, 103–104
  hello timer identifying roles, 93
  HSRP roles, 90, **94–95**
  multicast, 141
  VRRP, 101

H

HDLC. *See* High-Level Data-Link Control
HDSL. *See* high-bit-rate DSL
headend, 292
Hello packet, 177, 198, 369
  EIGRPv6, 214
  neighbor states, 242
Hello protocol, 177, 234, 236, **240–241**
hello timer, 93, 172, 356
Hello/Dead interval, 241
Hellos, 172
HFC. *See* hybrid fiber-coaxial
high-bit-rate DSL (HDSL), 294
High-Level Data-Link Control (HDLC), 284, 288–289, **298–300**, 337, 338, 357, 371
  configuring, 298–300, *299*
  encapsulation, 289, 298–300, 372
  frame format, *299*
  leased lines, 286
  mismatched encapsulation, **307–308**
  mismatched IP addresses, **308–309**
  Network layer protocol identification, 289, 298
  Physical layer specifications, 296
hold timer, 93–94, 172
hop count
  EIGRP, **187–188**
    default, 171

  maximum, 368
  AS size limits, 171
hostname, 304
Hot Standby Router Protocol (HSRP), 86, 89–90, **90–100**, 120, 364, 365, 366
  configuring and verifying, **95–99**
  group roles, **94–95**
  interface swapping, *92*
  interface tracking, **95**, *96*
  load balancing, **100**
  position elections, 90, 95, 96, 98
  timers, **92–94**
  virtual MAC address, **91–92**
  VRRP comparison, **101**
HP Openview, 112
HSRP. *See* Hot Standby Router Protocol
Hughes, 290
hybrid fiber-coaxial (HFC), 289, 292
hybrid routing protocols, 170, 171
HyperTerminal, *56*

I

IARP. *See* Inverse ARP
IBM, 288
ICMP echo, 129
ICMPv6, **141**
IDSL. *See* ISDN DSL
IEEE 802.3ad, 28
ietf, 312
IETF encapsulation, 312, 324, 337, 372
ifconfig, 130
IFS. *See* Cisco IOS File System
implicit deny, 138
INCMP message, 356
incomplete message, 356
INIT state, 242
input errors, 135
input queue drops, 135
install license, 74
int s0/0.*subinterface number*, 318
Integrated Services Digital Network (ISDN), 286–287, 288, 296, 300
inter-area routes, 239
interface, 153, 299, 357
  GRE tunnels, 333, 335
interface configuration mode, 178, 214, 263
interface errors, 134, 366
interface mode, 25
interface port=channel, 29, 30
interface tracking, **95**, *96*
interface tunnel *number*, 338
interface-dlci, 327
interfaces
  EIGRP metrics, 185–187
  EIGRP traffic control, 184–185
  Frame Relay
    single interface, **317–318**
    subinterfaces, **318–320**
  HSRP
    swapping, *92*
    tracking, **95**, *96*
  logical, 197
  passive, 185, 207, 208, 260, 356, 368
  serial, 288, 298, 307
    Frame Relay, 313
    HDLC, 298
    LAN technologies, 371
    OSPF load balancing, 261–262
    passive, 260
    WAN encapsulations, 288, 307
  split horizon, **195–197**
  trunk, 155
  tunnel, 333–334
internal routers, 234, 238, 268
Internet Protocol Control Protocol (IPCP), 301
Internet service provider (ISP), 285, 328
Internetwork Operating System (IOS), 48, 354–355
  backing up, **60**, 74, 363
  backing up and restoring, **57–68**
  backup to TFTP, 57–58, 60
  boot sequence, 50
  licensing, **68–74**
  mini-IOS, 49, 362

  - restoring, **61**, **63**, 74
  - upgrading, **61**, **63**, 74
  - upgrading with IFS, **65–68**
  - WAN support, 284, **288–290**
- Inter-Switch Link routing (ISL), 158, 159
- Inverse ARP (IARP), 313, 371, 372
- IOS. *See* Internetwork Operating System
- ip access-list, 115
- ip address, 299, 317
  - GRE tunnels, 333, 334, 335
- IP addresses
  - global, 140, 366
    - LMI, 314
  - mismatched, **308–309**
  - resolution states, 145–146
- IP Base licensing, 69
- ip flow egress, 118
- ip flow ingress, 118
- ip flow-export, 118
- ip host, 139
- IP network connectivity
  - troubleshooting, **128–140**, 163, 366
    - ACLs, 138–139
    - default gateway, 128–133
    - HTTP requests, 139–140
    - interface statistics, 134–135
    - local host connectivity, 130–132
    - local LAN, 132–133
    - pinging, 129
    - router configurations, 135–137
    - server configuration, 133–134
    - steps for, 129–130
    - telnet, 137–138, 139
  - VPNs, 328
- ip ospf network point-to-multipoint, 325
- ip route, 137
- IP routing, 128–129
- ipconfig, 130, 133, 146, 292
- IPCP. *See* Internet Protocol Control Protocol
- IPsec, **330**, 357, 373
  - transforms, **331–332**
- IPv6, 178
  - addresses, 141
  - ARP cache verification, 163
  - default gateway, 366
  - EIGRPv6, **214–219**
  - troubleshooting network connectivity, **140–149**, 163
    - ACLs, 148–149
    - address resolution states, 145–146
    - cables, 142
    - default gateway, 141–142, 146–147
    - ICMPv6, **141**
    - local host connectivity, 143
    - local LAN, 143–144
    - NDP, **141–142**
    - routing tables, 147–148
    - steps, 142–143
- ipv6 eigrp, 215
- ipv6 ospf, 264, 356
- ipv6 ospf *process-ID* area, 268
- ipv6 route, 148
- ipv6 router eigrp, 214, 216
- ipv6 router eigrp as, 356, 368
- ipv6 router ospf, 263
- ipv6 unicast-routing, 214
- IPX, 178
- ISDN. *See* Integrated Services Digital Network
- ISDN DSL (IDSL), 294
- ISL. *See* Inter-Switch Link routing
- ISP. *See* Internet service provider
- ISR series routers
  - password recovery, *55*, *56*
  - removable memory, 68
  - software licenses, 68, 362
- ISRG2 routers, 48
- ITU-T, 315

J

- J0, 286

K

K values, 172, 203, 368
  EIGRP troubleshooting, 204, 207–210, 213
  EIGRPv6 troubleshooting, 217
  formula for uses, 186
  metric association, 186–187, *187*
  mismatched, 204, 208, 213
keepalives, 314

L

L2F. *See* Layer 2 Forwarding
L2TP. *See* Layer 2 Tunneling Protocol
LACP. *See* Link Aggregation Control Protocol
LAN. *See* local area network
LAPB. *See* Link Access Procedure, Balanced
Layer 2 Forwarding (L2F), 330
Layer 2 Tunneling Protocol (L2TP), 330
layer 2 WAN encapsulation, 288
layer 2 WAN technologies, 371
LCP. *See* Link Control Protocol
learning state, 5, 13
leased lines, 286, 296
license boot module, 71, 74
License boot module, 363
license clear, 74
license install, 69–70, 74
license install url, 362
license save, 73
license save flash, 363
licenses, 48, 74, 362
  backing up and uninstalling, **73–74**
  Data, 69
  installing, 74
  IP Base, 69
  ISR series routers, 68, 362
  permanent, 71, 74
  removing, 363
  RTU, 68, **70–73**, 363
  Security, 69
  Unified Communications, 69
  verifying, 76
licensing, **68–74**
Link Access Procedure, Balanced (LAPB), 289
Link Aggregation Control Protocol (LACP), 28–30, 360, 361
Link Control Protocol (LCP), 300, 338, 357
  configuration options, **301–302**
link costs, 13, 370
  OSPF troubleshooting, 256
  STP, **6–7**
link-establishment phase, 302
link-local, 140, 219, 263
Link-State Advertisements (LSAs), 234, 236, **238–240**, 356, 370
  ASBRs, 240
  flooding zones, 238
  neighbor states, 243
  Type 1, 238–239, *239*
  Type 2, 239
  Type 3, 239
  Type 4, 240
  Type 5, 240
  types, 238–240, *240*, 268
link-state database (LSDB), 234, 238, 241, 246
  information exchange, 243
  synchronization, 236
link-state packets, 170
link-state protocols, 170
Link-State Request (LSR), 236, 243
Link-State Update (LSU), 236, 243
listening state, *5*
LMI. *See* Local Management Interface
LMI DLCI, 322–323
load balancing, 120, 366. *See also* Gateway Load Balancing Protocol
  EIGRP route selection, 189, 191–192, 368
  equal-cost, 200–201, 368
  HSRP, **100**
  OSPF troubleshooting, 261–262
  port state transitioning, 361
  unequal-cost, **193–195**, 201, 367, 368
  WANs, 190

load sharing, 103
LOADING state, 243, 370
local area network (LAN), 284
local loop, 285
Local Management Interface (LMI), 313, **314–315**, 318
  DLCIs, 323
  information communicated, 314
  message formats, 315, 323
  traffic statistics, 320–321
logging, 109, 111, 355
logging buffer, 107
logging buffered, 109
logging console, 109
logging host *ip_address*, 111, 365
logging trap, 111, 120, 364
logging trap warnings, 111
logical interfaces, 197
loopback address, 131
LRE. *See* Cisco Long Range Ethernet
LSAs. *See* Link-State Advertisements
LSDB. *See* link-state database
LSR. *See* Link-State Request
LSU. *See* Link-State Update

M

MAC address table, 22, 150–153
  thrashing, 2
MAC addresses, 129
  HSRP, 91, 92
  Proxy ARP, 86–87
  virtual, 89, **91–92**
  VLAN troubleshooting, 150–153
MAN. *See* metropolitan area network
management information base (MIB), **113–114**
maximum burst rate (MBR), 311
maximum hop count, 368
maximum paths
  EIGRP, **187–188**
  unequal-cost load balancing, 194, 367
maximum transmission unit (MTU), 186, 289
maximum-paths, 187
MBR. *See* maximum burst rate
MD5, 303
metric maximum-hops, 187
metric weights, 209
MetroEthernet, 284, 290
metropolitan area network (MAN), 290
MIB. *See* management information base
mini-IOS, 49, 362
MPLS. *See* MultiProtocol Label Switching
MTU. *See* maximum transmission unit
multi-area configurations, 234
  areas, 237–238
  basic, **243–245**
  router roles, 237–238, 268
multi-area OSPF networks, *235*, 235–236, 356, **369–371**
  adjacency requirements, **236–237**, 369, 370
  component categories, **236–243**
  configuration, **243–245**
  Hello protocol, **240–241**
  LSAs, **238–240**
  monitoring, 268
  neighbor states, **241–243**
  router roles, **237–238**
  scalability issues, 267
  troubleshooting, **245–262**, 268
    access control list, 257
    adjacencies, 254
    EIGRP, 258–260
    link costs, 256
    load balancing, 261–262
    ping, 255
    Ping, 254–255, 258
    routing protocols, 257
    show ip interface, 255–257
    show ip interface brief, 255
    show ip ospf interface, 254
    show ip ospf neighbor, 254, 258
    show ip protocols, 257, 258, 260
    show ip route, 254, 259
  verifying, *245*, **245–253**
    Ping, 246

multicast groups, 141
multicast traffic
  EIGRP, 176
  EIGRPv6, 214
  OSPFv3, 263
  reliable, 176
multicasting, 314
multiple frame copies, 2
multiplexing, 318
multipoint subinterfaces, 319–320
MultiProtocol Label Switching (MPLS), 101, 284, 287, 290

N

NA. *See* neighbor advertisement
name, 153
name resolution, 356
native VLAN, 159–162, 163
NBMA. *See* nonbroadcast multi-access
NCP. *See* Network Control Protocol
NDP. *See* neighbor discovery
neighbor, 185
neighbor advertisement (NA), 141, 142
neighbor database, 369
neighbor discovery (NDP)
  EIGRP, **171–176**
  troubleshooting network connectivity, **141–142**
neighbor resolution table, 145
neighbor solicitation (NS), 141, 142, 145, 356
neighbor states, **241–243**
neighbor table, 174, 178, 220
NetFlow, 86, **116–119**, 120, 355, 364, 365
  configuring, **118–119**
  flows, **117–118**
  overview, **117**
netsh interface ipv6 show neighbor, 146–147
network, 180
Network Control Protocol (NCP), 289, 300–302, 338, 357
  serial link troubleshooting, 305–306
network layer protocol phase, 302
network link advertisement (NLA), 239, 370
network loops, 2, 6
  PortFast configuration, 26
network management station (NMS), 86, 112, 120
network simulator, 377
network topology information
  ABR storing, 238
  DUAL, 177
  EIGRP route discovery and maintenance, 178
  EIGRP troubleshooting, 204
  synchronization, 170
next-hop attributes
  EIGRP configuration, 190
  EIGRP troubleshooting, 201
  IPv6, 148, 217
  OSPFv3, 263
NLA. *See* network link advertisement
NMS. *See* network management station
no auto-summary, 183–184, 211
no bandwidth, 191
no distribute-list, 213
no encapsulation, 307
no ip address, 357
no ip route, 137
no license boot module, 74
no logging buffered, 110
no logging console, 110
no passive-interface, 260
no router eigrp, 259
no service timestamps, 111
no shutdown, 214, 216, 220, 299
nonbroadcast multi-access (NBMA), 242, 309, 325, 372, 373
non-designated port, 4
non-root bridges, 4
nonvolatile RAM (NVRAM), 49, 50–51, 355, 362, 363
NS. *See* neighbor solicitation
NVRAM. *See* nonvolatile RAM

O

- OC. *See* Optical Carrier
- OC-3, 286
- OC-12, 286
- OC-48, 286
- OIDs. *See* organizational IDs
- Open Shortest Path First (OSPF), 2, 28, 128, 170, 180
  - Frame Relay routing problems, 325–326, *326*
  - Hello protocol, **240–241**
  - multi-area networks, *235*, 235–236, 356, **369–371**
    - adjacency requirements, **236–237**, 369, 370
    - component categories, **236–243**
    - configuration, **243–245**
    - Hello protocol, **240–241**
    - LSAs, **238–240**
    - monitoring, 268
    - neighbor states, **241–243**
    - router roles, **237–238**, 268
    - scalability issues, 267
    - troubleshooting, **245–262**, 268
    - verifying, *245*, **245–253**
  - RID, 370
  - routing table, 252
  - scalability, **234–236**, 267
  - single-area network, 235, *235*
  - topology changes, 236
  - troubleshooting, 268
- Optical Carrier (OC), 286
- organizational IDs (OIDs), 113
- organizationally unique identifier (OUI), 91
- OSPF. *See* Open Shortest Path First
- OSPFv2, 356
- OSPFv3, **262–267**, 356, 370
  - configuring, 263–265, 268
  - RID, 370
  - verifying, **265–267**
- OUI. *See* organizationally unique identifier
- output errors, 135
- output queue drops, 135

P

- packet switching, 287, 310
- PAgP. *See* Port Aggregation Protocol
- PAK. *See* product authorization key
- PAP. *See* Password Authentication Protocol
- passive interfaces, 185, 207, 208, 260, 356, 368
- passive state, 201
- passive-interface *interface-type interface-number*, 185
- Password Authentication Protocol (PAP), 302, 303
- passwords
  - encryption, 304
  - recovering, **55–57**, 74, 362
- path cost, 4
- permanent licenses, 71, 74
- permanent virtual circuits (PVCs), 312, 313, 321–322, 337, 371, 372
- Per-VLAN Spanning Tree+ (PVST+), 10, **11–12**, 32, 361
- PID. *See* product ID
- ping, 356
  - EIGRP
    - configuration, 183–184
    - troubleshooting and verifying, 204, 206, 210, 213
  - GRE tunnel verification, 336–337
  - IP connectivity troubleshooting, 129–134, 136–138
  - IPv6 connectivity troubleshooting, 143–145, 148
  - OSPF troubleshooting, 255
  - TFTP verification, 60
  - VLAN connectivity troubleshooting, 154, 161
- Ping
  - connectivity testing, 129, 130, 141
  - multi-area OSPF
    - troubleshooting, 254–255, 258
    - verification, 246
  - OSPFv3, 263

- plain old telephone service (POTS), 286, 288, 294
- point of presence (POP), 285
- point-to-multipoint links, 197
- point-to-point links, 288
  - elections, 246
- Point-to-Point Protocol (PPP), 284, 286, 289, **300–309**, 371, 372
  - authentication, *304*, 357, 371
    - configuring, **303–304**
    - debugging, **306**
    - failed, *305*
    - methods, **303**
  - components, 300–301, *301*
  - configuring on Cisco routers, **303**
  - Data Link layer protocols, 338
  - encapsulation, 300–306
    - mismatched, **307–308**
  - LCP configuration options, **301–302**
  - mismatched IP addresses, **308–309**
  - Physical layer specifications, 296
  - serial link verification and troubleshooting, **304–306**, 338
  - session establishment, **302**
  - verification, 338
- Point-to-Point Protocol over ATM (PPPoA), 294
- Point-to-Point Protocol over Ethernet (PPPoE), 284, 289, 357, 372
  - ADSL, **294–295**
- point-to-point subinterfaces, 319
- Point-to-Point Tunneling Protocol (PPTP), 330
- POP. *See* point of presence
- Port Aggregation Protocol (PAgP), 28, 29, 354, 360, 361
- port channeling, 28, 29, 361
- port channels, 28, 33
  - configuring and verifying, **29–32**
- port cost, 4
- port modes, 157–159, 366
- port states, 33
  - BPDU Guard, **26**
  - load balancing, 361
  - spanning-tree, **5–6**
  - STP failure, 22
- PortFast, **24–26**, 33, 360, 361
- ports
  - alternate, 5
  - backup, 5
  - blocked, 5
  - bridge
    - PAgP, 29
    - roles, 4–5
  - console
    - restoring IOS, 63
    - syslog messages, 107
  - designated, 4, 15–16
    - RSTP, 13
  - designated forwarding, 14, 18
  - disabled, 5
  - dynamic auto, 158
  - forwarding, 4
  - non-designated, 4
  - root, 4, 354
    - RSTP, 13, 14
  - trunk, 155, 156
  - trunking configurations, 157
- POST test, 355
- POTS. *See* plain old telephone service
- PPP. *See* Point-to-Point Protocol
- ppp authentication chap, 357
- ppp authentication chap pap, 304
- PPPoA. *See* Point-to-Point Protocol over ATM
- PPPoE. *See* Point-to-Point Protocol over Ethernet
- PPTP. *See* Point-to-Point Tunneling Protocol
- preemption, 103
- priority value, 9
- privileged mode, **56**
  - passwords, 57
- process ID, 257
- product authorization key (PAK), 69
- product ID (PID), 69, 74, 76
- Proxy Address Resolution Protocol (Proxy ARP), 86–87
- PuTTY, 56
- PVC status field, 322

PVCs. *See* permanent virtual circuits
PVST+. *See* Per-VLAN Spanning Tree+
pwd, 65

Q

Q value, 199
Q.933A LMI format, 315, 323
Query packet, 176

R

RA. *See* router advertisement
RADSL. *See* Rate Adaptive DSL
random access memory (RAM), 49, 50, 57, 59, 355
  neighbor and topology tables in, 174
Rapid PVST+, 10, 361
Rapid Spanning Tree Protocol (RSTP), 10, **12–16**, 32, 360, 361. *See also* 802.1w
  enabling, 21–22
Rate Adaptive DSL (RADSL), 294
REACH state, 366
read-only memory (ROM), 49, 355
redistribution, 215
reference bandwidth, 262, 371
reliable multicast, 176
Reliable Transport Protocol (RTP), 171, **176–177**
reload, 362
remote access VPNs, 329, 357
removable memory, 60
Reply packet, 176
reported distance. *See* advertised distance
Retransmission Time Out (RTO), 198
RFC 1661, 300
RFC1483 routing, 294
RID. *See* router ID
Right-To-Use licensing (RTU), 68, **70–73**, 74, 363
RIP, 2
  maximum hop count, 368
RIPng, 214
RIPv2, 170
Rivest, Shamir, and Adleman encryption (RSA), 332
RLA. *See* router link advertisement
ROAS. *See* router on a stick
ROM. *See* read-only memory
ROM monitor, 49
ROM monitor mode, 52
root bridge, 3, 354, 361
  configuring, 20
  elections, 8–10
  per-VLAN, 11, 360, 361
  RSTP, 13–15
  selecting, **9**, 11
root port, 4, 354
  RSTP, 13, 14
route discovery, EIGRP, **178**
route maintenance, EIGRP, **178**
route print, 130
route redistribution, 238
route selection
  EIGRP, **188–193**
  load balancing, 189, 191–192, 368
  routing table, 188–189
  topology table, 189–190, 191
route table, 178
router advertisement (RA), 141, 142
router configuration mode, 178, 263
router eigrp, 179–181, 183, 187, 209, 211, 213
router ID (RID), 215, 239, 367
  EIGRP, 244
  OSPF, 244, 370
  OSPF Hello protocol, 240
  OSPFv2, 262
  OSPFv3, 262–263, 264
router link advertisement (RLA), 238
router on a stick (ROAS), 100
router ospf, 243, 245, 254–255
router ospf process-id, 244
router solicitation (RS), 141, 142
router-id, 216, 243, 263

- routers
  - ABRs, 234, 237–238, 240, 268
  - active, 90, 94, 95
  - ASBRs, 234, 237, 238, 240, 268, 369
  - AVR, 365
  - backbone, 234, 237, 268
  - boot sequence, **49–50**, 74
  - designated, 239, 246, 249, 325, 370
  - internal, 234, 238, 268
  - internal components, **48–49**, 74
  - memory, 355
  - multi-area configurations, 234
    - roles, **237–238**, 268
  - password recovery, **55–57**
  - priority, 241
  - reloading, **56**, 57
  - serial connectors, 296
  - standby, 90, 93–94, 95
  - statistics, 163
  - successor, 174
  - virtual, 89, 90, 94–95, 103, 365
    - gateways, 101
    - GLBP, 103
    - HRSP, 90, 94–95
- routing
  - Frame Relay troubleshooting, *325*, 325–327, *326*
  - ISL, 158
- routing database
  - EIGRP
    - corrupted, 176
    - synchronizing, 176
    - troubleshooting, 200
  - LSDB, 234, 238, 241, 246
    - information exchange, 243
    - synchronization, 236
  - multi-area OSPF, 234
    - monitoring, 268
    - synchronization, 236
- routing protocols, 2
  - advanced distance-vector, 170, 171
  - classless, 171, 181
  - distance-vector, 171
  - hybrid, 170, 171
  - link-state, 170
  - OSPF troubleshooting, 257
- routing table, 212
  - ACL problems, 213
  - EIGRP route selection, 188–189
  - EIGRP troubleshooting, 200
  - EIGRPv6, 218–219
  - mismatched IP addresses, 309
  - OSPF, 252
  - topology table population, 174
  - troubleshooting, 135–137
- RS. *See* router solicitation
- RSA. *See* Rivest, Shamir, and Adleman encryption
- RSTP. *See* Rapid Spanning Tree Protocol
- RTO. *See* Retransmission Time Out
- RTP. *See* Reliable Transport Protocol
- RTU. *See* Right-To-Use licensing
- RXBOOT, 49

S

- satellite, 290
- SDLC. *See* Synchronous Data Link Control
- SDSL. *See* Synchronous DSL
- Secure Sockets Layer (SSL), 329, 372
- SecureCRT, 56
- security gateway, 332
- Security license, 69
- Sequence Number field, 331
- sequence numbers, 111, 174
- serial encapsulation, 300, 337, 372
  - troubleshooting, 324–325, *325*, *326*
- serial interfaces
  - Frame Relay, 313
  - HDLC, 298
  - LAN technologies, 371
  - OSPF load balancing, 261–262
  - passive, 260
  - WAN encapsulations, 288
    - mismatched, 307
- serial lines, 286
  - NBMA networks, 325

serial links
  point-to-point, 266
  verifying and troubleshooting, **304–306**, 338
serial transmission, **296**
service password-encryption, 304
service sequence-numbers, 111
service timestamps log datetime msec, 111
SET messages, 113
setup mode, 362
severity levels, 108–109, 111, 120
shortest path first (SPF), 234
show access-lists, 138, 213
show cdp neighbors, 18
show cdp neighbors detail, 309
show controllers *interface*, 297–298
show dtp interface *interface*, 155, 157, 158, 366
show etherchannel port-channel, 31, 354
show etherchannel summary, 32
show file, 66
show file information, 67
show flash, 59–60, 66, 362
show frame ?, 320
show frame map, **323**
show frame pvc, **321–322**
show frame-relay ?, 371
show frame-relay lmi, **320–321**
show frame-relay map, 316
show frame-relay pvc, 317
show glbp, 106–107, 120, 355, 365
show glbp brief, 106–107, 365
show interface, 134, 209, 299, 300, 357
  Frame Relay monitoring, **322–323**
  mismatched encapsulations, 307
  mismatched IP addresses, 308
  PPP troubleshooting, 305
show interface *interface* switchport, 158, 366
show interface trunk, 159, 160, 162, 366
show interface tunnel, 338
show interfaces, 190, 207, 372
show interfaces *interface* switchport, 150, 151, 155, 156, 161
show interfaces *interface* trunk, 155, 156
show interfaces trunk, 154, 155
show interfaces tunnel 0, 336
show ip cache flow, 119, 355, 364
show ip eigrp events, 203
show ip eigrp interface detail *interface*, 199
show ip eigrp interfaces, *198*, 199, 204, 205
show ip eigrp neighbors, 198, *198*, 204, 209, 210, 220, 368
show ip eigrp topology, 189, 192, *198*, 200–201, 202, 204, 205
show ip eigrp topology | section *network*, 191
show ip eigrp traffic, *198*, 202
show ip flow export, 119
show ip flow interface, 118–119
show ip interface, 138, 195
  OSPF troubleshooting, 255–257
show ip interface brief, 136, 144–145, 205, 206
  GRE tunnel verification, 335
  OSPF troubleshooting, 255
show ip interfaces brief, 309
show ip ospf, **247–248**, 268
show ip ospf database, *245*, **252–253**, 268
show ip ospf interface, *245*, **248–249**, 254, 370
  OSPF troubleshooting, 254
show ip ospf neighbor, *245*, 246, 268, 370
  OSPF troubleshooting, 254, 258
show ip protocols
  EIGRP
    configuring, 186, 193
    troubleshooting, 198, 202, 206, 208, 211
  EIGRPv6, 217
  K values, 186
  OSPF troubleshooting, *245*, **250–251**, 254, 257, 258, 260, 262, 356
  unequal-cost load balancing, 193
show ip route
  EIGRP auto-summarization, 181–184
  EIGRP route selection, 191
  EIGRP troubleshooting, 200, 210, 212, 213

  - GRE tunnel verification, 336
  - IP network connectivity troubleshooting, 135–137
  - multi-area OSPF troubleshooting, *245*, **251–252**
  - OSPF troubleshooting, 254, 259
  - unequal-cost load balancing, 195
- show ip route eigrp, *198*, 199
- show ip route *network*, 191
- show ip route ospf, 252, 261–262, 268
- show ipv6 access-lists, 149
- show ipv6 eigrp interfaces, 216
- show ipv6 eigrp neighbors, 217
- show ipv6 interface brief, 145
- show ipv6 neighbors, 145, 163, 366
- show ipv6 ospf, 266–267
- show ipv6 ospf interface, 266
- show ipv6 ospf neighbor, 266
- show ipv6 protocols, 217, 265
- show ipv6 route, 147–148
- show ipv6 route eigrp, 218–219
- show ipv6 route ospf, 265
- show ipv6 topology, 217–218
- show license, 71, 74, 76, 355, 363
- show license feature, 72, 74, 76, 363
- show license udi, 69, 74, 76, 363
- show logging, 110
- show mac address-table, 150–153
- show protocols, 205, 206
- show run, 212, 304
- show run | begin interface, 300
- show running-config, 300, 309
- show spanning-tree, 17, 19, 33, 354, 360
- show spanning-tree summary, 21–22, 354
- show standby, 98, 120, 355
- show standby brief, 98–99
- show version, *52–53*, *59–60*, 73, 74, 362, 363
- show vlan, 150, 154, 155
- show vlan brief, 150, 152, 155, 156
- Simple Network Management Protocol (SNMP), 86, **112–116**, 120, 355, 363, 364
  - ACL configuration, 116
  - configuring, **114–116**
  - MIB, **113–114**
  - versions, 113
- single-area OSPF networks, 235, *235*
- site-to-site VPNs, 330, 357
- SLA. *See* summary link advertisement
- smart-serial, 296
- smooth round-trip timer (SRTT), 172, 198
- SNMP. *See* Simple Network Management Protocol
- snmp-server, 114–116, 363
- snmp-server community, 115–116
- SNMPv3, 113, 366
- software configuration bit meanings, *51–52*, *52*
- Spanning Tree Protocol (STP), **2–9**, 32, 33, 354, 361, 362. *See also* Rapid Spanning Tree Protocol
  - bridge port roles, **4–5**
  - convergence, **6**, 360
  - failure consequences, **22–24**
  - failure types, 22
  - link costs, **6–7**
  - operations, **7–9**
  - port states, **5–6**
  - root bridge selection, **9**
  - states, 33
  - terms, **3–7**
  - types, **9–16**
  - verifying, 17
- spanning-tree, 20
- spanning-tree algorithm (STA), 2, 4
- spanning-tree bpduguard enable, 26
- spanning-tree mode rapid-pvst, 21–22
- spanning-tree port states, 5–6
- spanning-tree portfast, 25, 354
- spanning-tree portfast bpduguard, 26
- spanning-tree vlan *number* root primary, 354
- speed settings, 134–135
- SPF. *See* shortest path first
- split horizon, **195–197**, *196*, 318, 371
- SRTT. *See* smooth round-trip timer
- SSL. *See* Secure Sockets Layer
- STA. *See* spanning-tree algorithm
- STALE state, 366

standby *group* ip *virtual_ip*, 96–98
standby groups, 90
standby router, 90, *92*, 93–94, *95*
standby timer, 94
STP. *See* Spanning Tree Protocol
study tools, 377–378
subinterfaces, 197, **318–320**, 371
  configuring, 318–319
  Frame Relay troubleshooting, 327–328
  types, 319–320
subnet mask, 170
subnets
  EIGRP, 182, 327
    advertising, 183–184, 185
    configuring, 182–184
  Frame Relay troubleshooting, 327
  VLSMs, 170, 177
    EIGRP, **181–184**
successor route, 175, 367, 368
successor router, 174
summaries, 171
summarization, 170
  EIGRP, **181–184**
summary link advertisement (SLA), 239, 370
SVCs. *See* switched virtual circuits
switch access vlan, 153
switch native vlan, 356
switched networks
  bridge ID, 9
  CST, 10
  layer 2, 149
  PVST+, 11
  RSTP, 12
  STP, 2, 3, 33
switched virtual circuits (SVCs), 312
switches
  CPU load, 23–24
  statistics, 163
switching loops, 2, 3, 360
switchport access vlan, 149, 150, 153
switchport mode, 155, 157
switchport mode dynamic, 155, 158
switchport mode dynamic desirable, 158, 159
switchport mode trunk, 30, 31
switchport native vlan *vlan*, 163
switchport trunk allowed vlan, 31
switchport trunk encapsulation ?, 160
switchport trunk encapsulation dot1q, 30, 160
switchport trunk native vlan *vlan*, 155, 159, 162
symmetric encryption, 332
symmetrical DSL, 293
Synchronous Data Link Control (SDLC), 288, 298
Synchronous DSL (SDSL), 294
syslog, 86, **107–112**
  configuring and verifying, **109–112**
  message format, 108
  message gathering, 107
  severity levels, *108*, 108–109, 111, 120, 364
syslog server, 107, 111

T

T1, 286, 313
T3, 286, 288
TAC. *See* Cisco Technical Assistance Center
T-carrier transmission, 286
telnet, 137–139
Telnet, 117
terminal lines, 107
terminal monitor, 107
test engine, 376
test environments, 371
TFTP, 50
  booting from, *53–55*
  IOS backup, *57–58*, 60
tftp-server flash:ios_name, 61
3DES. *See* Triple DES
timers
  active, 94
  dead, 356
  hello, 93, 172, 356
  hold, 93–94, 172
  HSRP, **92–94**
  SRTT, 172, 198
  standby, 94

Token Ring, 371
toll network, 285
topology database, 252
topology table, 174, 178, 200, 220
  EIGRP route selection, 189–190, 191
  FS, 175, 368
  successor routes in, 368
ToS. *See* type of service
traceroute, 132, 144, 254
traffic balancing, 103
  per-host, **104**
traffic flow confidentiality, 331, 332
trap levels, 364, 365, 366
TRAP messages, *112*
Triple DES (3DES), 331, 332
troubleshooting, 128
  adjacencies, 368, 369
  EIGRP, **197–214**
  Frame Relay networks, **324–328**, 338
    congestion control, **316–317**
  IP network connectivity, **128–140**, 163
  IPv6 network connectivity, **140–149**, 163
  multi-area OSPF networks, **245–262**, 268
  OSPF, 268
  PPP, **304–306**, 338
  serial links, **304–306**, 338
  steps for, 129–130
  study tools, 377
  trunk, 154–162
  VLANs, **149–162**
    scenario, **150–154**
    steps for, **149–150**
    trunk, **154–162**
trunk encapsulation, 159–162
trunk interfaces, 155
trunk lines, 285
trunk ports, 155, 156
trunk troubleshooting, **154–162**
trunking, 30, 156. *See also* Dynamic Trunking Protocol
  port configurations, 157
  port modes, 157–159
tunnel destination, 335
tunnel mode, 334, 338
tunnel mode gre, 334
tunnel mode gre ip, 334
tunnel source, 334, 335
tunneling protocols, 330, 373
  GRE, **332–337**
  interface, 333–334
  verifying, 336
2WAY state, 241–242
type of service (ToS), 210

U

UDI. *See* unique device identifier
UDP. *See* User Datagram Protocol
unequal-cost load balancing, **193–195**, 201, 367, 368
Unified Communications license, 69
unique device identifier (UDI), 69, 74, 76, 354
universal image, 68
untagged frames, 159, 163
Update packet, 176
User Datagram Protocol (UDP), 103
user mode, passwords, 57
username, 304
username *username* password *password*, 357

V

V.24, 300
V.35, 296, 297, 300
variable length subnet masks (VLSMs), 170, 177
  EIGRP, **181–184**
variance parameter, 194
VDSL. *See* very-high-data-rate DSL
verification
  adjacency, 254
  bridge ID, **17–22**
  EIGRP, **197–214**, 220
  EIGRPv6, 217–219
  flash memory, **59–60**
  Frame Relay, 338
  GRE tunnels, **335–337**, 338
  HSRP, **95–99**

- multi-area OSPF networks, *245*, **245–253**
- OSPFv3, **265–267**
- port channels, **29–32**
- PPP, 338
- serial links, **304–306**, 338
- spanning trees, 17
- syslog, **109–112**

Very Small Aperture Terminal (VSAT), 290
very-high-data-rate DSL (VDSL), 294
via Connected, 218
virtual circuits, **312**
- DLCI status, 314
- LMI information, 314–315
- PVCs, 312, 313, 321–322, 337, 371, 372
- SVCs, 312

virtual gateway, 103
virtual LANs (VLANs). *See also* Per-VLAN Spanning Tree+
- HSRP load balancing, 100
- native, 159–162, 163
- root bridges, 11, 360, 361
- troubleshooting connectivity, 128, **149–162**, 356
  - scenario, **150–154**
  - steps for, **149–150**
  - trunk, **154–162**
- VPN similarities, 328

virtual MAC address, 89, **91–92**
virtual private dial-up networks (VPDNs), 330
virtual private networks (VPNs), 101, 284, **328–332**, 357, 372, 373
- benefits, **329–330**
- encryption, **332**
- extranet, 330, 357
- IP network connectivity, 328
- IPsec, **331**
- IPsec transforms, **331–332**
- remote access, 329, 357
- site-to-site, 330, 357
- VLAN similarities, 328

virtual router, 89, *365*
- gateways, 101
- GLBP, 103
- HRSP, 90, 94–95

Virtual Router Redundancy Protocol (VRRP), 86, 90, **101**, 120
vlan, 153
VLAN database, 150
VLAN ID, 11, 361
VLAN leaking, 154
VLANs. *See* virtual LANs
VLSMs. *See* variable length subnet masks
VPDNs. *See* virtual private dial-up networks
VPNs. *See* virtual private networks
VRRP. *See* Virtual Router Redundancy Protocol
VSAT. *See* Very Small Aperture Terminal

W

WAN. *See* wide area network
Web VPN, 329
wide area network (WAN), **284**, 357, **371–373**. *See also* Frame Relay
- connection bandwidth, **286**
- connection types, **286–287**, *287*
- DTE/DCE, **297–298**
- encapsulation
  - layer 2, 288
  - mismatched, **307–308**
- IOS support, **288–290**
- load balancing, 190
- serial connections, **296–298**
- serial connectors, 296
- serial transmission, 296
- terms, **285**
- troubleshooting, 128

wildcards, 180

X

X.25, 287, 288
xDSL, 289, *293*, 294
Xmodem, 63

SPECIAL OFFER FROM TODD LAMMLE*

Save 10% on CCNA Video Training from Todd Lammle

Watch online, on your smartphone, or on your tablet

Your 30-Day subscription to the video training includes access to all 14 CCNA Training Modules:

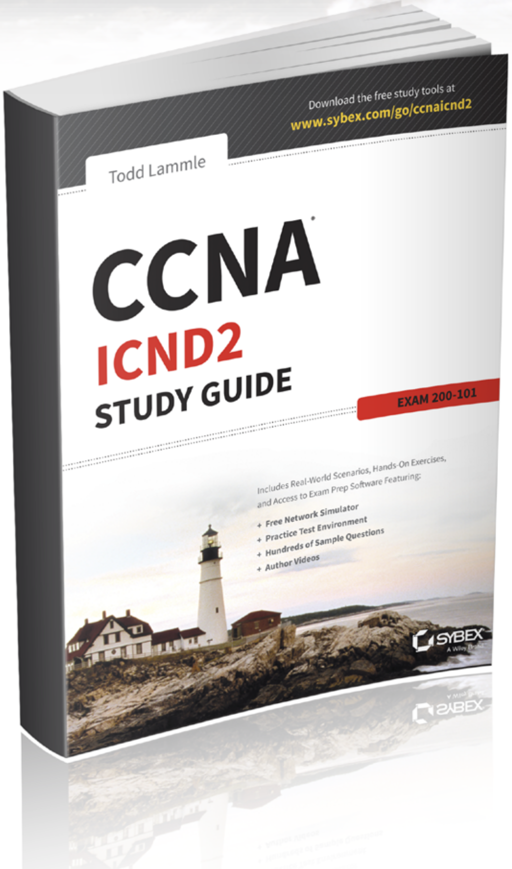
- Module 1: Internetworking
- Module 2: Introduction to TCP/IP
- Module 3: IP address, Subnetting, VLSM and Summarization
- Module 4: Introduction to the Cisco® IOS
- Module 5: IOS Management and IP Troubleshooting
- Module 6: IP Routing
- Module 7: EIGRP and OSPF
- Module 8: Layer-2 Switching and Spanning Tree Protocol
- Module 9: Virtual LANs (VLANs)
- Module 10: Access-lists
- Module 11: Network Address Translation
- Module 12: Wireless LANs
- Module 13: IPv6
- Module 14: Wide Area Networks (WANs) and Security Device Manager

Get details at http://www.lammle.com/online-ccna-series.

Use Promo Code **SYBEX**

**This offer will be fulfilled and customer service will be provided by Todd Lammle. John Wiley & Sons, Inc. is not affiliated with www.lammle.com and the products offered there.*

FREE Sybex Online Study Tools



Download the free study tools at
www.sybex.com/go/ccnaicnd2

Todd Lammle

CCNA
ICND2
STUDY GUIDE

EXAM 200-101

Includes Real-World Scenarios, Hands-On Exercises, and Access to Exam Prep Software Featuring:

- + Free Network Simulator
- + Practice Test Environment
- + Hundreds of Sample Questions
- + Author Videos

SYBEX
A Wiley Brand

Along with this Study Guide you get access to a full set of test prep tools to help you thoroughly prepare for the exam, including:

- + **Custom practice test environment** with scoring and instant feedback
- + **Hundreds of sample questions** to help you assess your knowledge of the content
- + **Electronic flashcards** for quick testing of key topics and terms
- + **Author Videos** covering key Cisco networking concepts
- + **A Network Simulator** to perform all hands-on labs in the book
- + **Glossary of Key Terms** that you'll need to know for the exam

Get access to the study tools from the Downloads tab on the product page at www.sybex.com/go/ccnaicnd2.

SYBEX®
A Wiley Brand